1990年7月1日発行（毎月1回1日発行）第9巻7号通巻99号　昭和58年11月2日第三種郵便物認可

特集 マシン語への第一歩

S-OSリロケータブルアセンブラWZD
ハードウェア工作入門/PurePASCAL
ノーマルX1対応コマンドシェルINTEGRAL X1

7
1990

オー！エックス
定価560円

SHARP

X68000

SX-WINDOW

ひらかれた知性。

見はてぬ夢の象徴。

次代のインテリジェンス、"SX-WINDOW"※搭載。

※SX-WINDOWの起動には、メインメモリ2MBが必要です。PROIIシリーズ（CZ-653C/663C）でSX-WINDOWをご使用の際には、あらかじめ別売の1MB増設RAMボードCZ-6BE1Bを増設してください。

資料請求券 X68000

●いま、こだわり続けてきたある執着がまさに帰結しようとしています。グラフィカルユーザーインターフェイス"SX-WINDOW ver.1.0"。もちろん、X68000には発売当初よりビジュアルシェルが搭載されていたことはご存じのとおりですが、クオリティグラフィックやマルチメディア、マルチタスク対応など真の意味での汎用性を志向した開発コンセプトからは、私たち自身ものたりなさを禁じ得なかったことも事実です。しかし、キャラクタユーザーインターフェイス全盛のその時代に、デスクトップの概念をいち早く採り入れた先見性は、現在のインターフェイスの在り方に対する的確な予測に基づくもので、何よりも、トレンディなユーザーの圧倒的な支持によって証明されています。パーソナルコンピュータがその意味どおり、個人のためのツールなら、インターフェイスの発展は必然です。このウィンドウシステムは、私たちX68000開発プロジェクトに携わったすべてのスタッフの指標であり、義務でもあったのです――。

●ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプのデスクトップ(テキスト面/単色4階調+カラー4色、グラフィック面/カラー65,536色中16色)、新感覚スクロールバー…こだわりの美学で高められたユーザーインターフェイス。イベント・ドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単におこなえる多機能コントロールパネルを搭載した本格ウィンドウシステムです。

●"SX-WINDOW"。このひらかれた知性は、今もそしてすぐ後に続く時代をも包含した質の高い「愉しみ」を提供するインターフェイスです。フレンドリーOS Human68kは、ここに、当初の目的の成就と共に、将来へ確かな展望を明示したといえるでしょう。さまざまなジャンルへ、拡がりと密度を高めるアプリケーション環境、インテリジェントなペリフェラル環境。こうしたトレンドを背景に、いま第4世代のX68000がデビューします。

NEW X68000
PERSONAL WORKSTATION

SUPER・EXPERT・PRO

ザ・ワークステーション。80MバイトハードディスクSCSIインターフェイスを標準装備。
SUPER HD 本体+キーボード+マウス・トラックボール
HDタイプ CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)〈6月発売予定〉

アートの系譜。**EXPERT II** 本体+キーボード+マウス・トラックボール
CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンダード。**PRO II** 本体+キーボード+マウス
CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)
HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)

15型カラーディスプレイテレビ(ドットピッチ0.39mm) CZ-602D-BK(ブラック)・-GY(グレー)……標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm) CZ-605D-BK(ブラック)・-GY(グレー)……標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm) CZ-613D-TN(チタンブラック)・-BK(ブラック)・-GY(グレー)……標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm) CZ-603D-BK(ブラック)・-GY(グレー)……標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm) CZ-604D-BK(ブラック)・-GY(グレー)……標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)

今回のゲームソフトウイルスの件では、ユーザー各位にご心配をおかけ致しました。ワクチンソフトに関しては、お近くの弊社サービスセンターにお尋ね下さい。X68000及び関係商品は、これまで通り安心してご愛用頂けます。

EXEリーダーズグッズ
プレゼント実施中

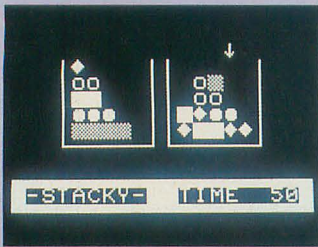
●いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズグッズをプレゼントします。詳しくはEXEショップにお問い合わせください。
●また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

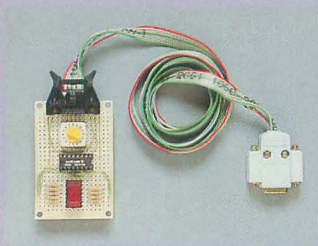
●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

シャープ株式会社

マイクロコンピュータショウ'90

DōGA・CGアニメーション講座

スタッキー

サーク

ダウンタウン熱血物語

ハードウェア工作入門

Oh!X

CONT

●特集

〈スタッフ〉
●編集長／前田 徹 ●編集／植木章夫 岡崎栄子 浅井研二 ●協力／有田隆也 中森 章 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／千野延明 織田洋子

表紙絵：塚田　哲也

ENTS

1990 JUL.
7

■広告目次

SHARP

クリエイティブマインドあふれる周辺機器が

ディスプレイ関連

カラーディスプレイテレビ

15型カラーディスプレイテレビ
CZ-602D-BK・-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格 115,000円(税別)
(スピーカー2個・チルトスタンド同梱)

NEW

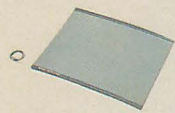
15型カラーディスプレイテレビ
CZ-613D-TN・-BK・-GY
標準価格 135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

カラーディスプレイ

14型カラーディスプレイ
CZ-603D-BK・-GY
標準価格 84,800円(税別)
(チルトスタンド同梱)

14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

21型カラーディスプレイ
CU-21HD
標準価格 148,000円(税別)
(スピーカー2個同梱)

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14/15型用)

チューナー

RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格 33,100円(税別)
(リモコン付)

アートツール

画像入力

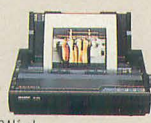
カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円(税別)

スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円(税別)

映像入力

カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

プリンタ

カラープリンタ

24ドット
熱転写カラー漢字プリンタ
CZ-8PC3
標準価格 65,800円(税別)
(信号ケーブル同梱)

48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)

カラービデオプリンタ

カラービデオプリンタ
CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット

カラーイメージジェット※3
IO-735X
標準価格248,000円(税別)
(信号ケーブル別売)

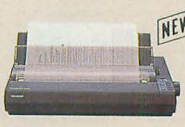
ドットプリンタ

NEW

24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)

NEW

24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)

NEW

24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

ファイル

ハードディスク

ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)

増設用ハードディスク
ドライブ(40MB)
(CZ-602C/652C/603C/653C内蔵用)
CZ-64H
標準価格 120,000円(税別)
(取付費別)
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。
※2 CZ-603D/604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。
※3 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。

X1・X1turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21HD	148,000円

映像・画像入力編集装置		
●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円
●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円

FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー(2本1組)標準装備、ミュージックツール同梱

プリンタ		
●24ピンカラー漢字プリンタ(80桁)	CZ-8PG1	130,000円
●24ピンカラー漢字プリンタ(136桁)	CZ-8PG2	160,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK10	97,800円
●24ドット熱転写カラー漢字プリンタ	CZ-8PC3	65,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4	99,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4-GY	99,800円
●カラービデオプリンタ	CZ-6PV1	198,000円
●カラーイメージジェット	IO-735X	248,000円

ファイル		
●ミニフロッピーディスクユニット(2HD・2D)※3	★CZ-520F	118,000円

シャープ株式会社
●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表

資料請求券 X68000ペリフェラル Oh!X 7月

X68000をサポート。

シャープペリフェラルファミリー
X68000

ボード

拡張メモリ

1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)

1MB増設RAMボード
(CZ-601C/611C/652C/653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)

2MB増設RAMボード ※4
CZ-6BE2
標準価格 79,800円(税別)

4MB増設RAMボード ※4
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)

GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX

FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI

MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム

モデムユニット ※5
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード

NEW

LANボード
CZ-6BL1
標準価格 268,000円(税別)
CZ-6BL2
標準価格 298,000円(税別)
※電源ユニット・ソフトウェア
(ネットワークドライバVer1.0)同梱

入力

インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)

マウス・トラックボール
CZ-8NM3
標準価格 9,800円(税別)

トラックボール
CZ-8NT1
標準価格 13,800円(税別)

マウス
CZ-8NM2A
標準価格 6,800円(税別)

ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット

拡張I/Oボックス(4スロット)
(CZ-600C/601C/611C/602C/612C/603C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー

アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック

システムラック
(CZ-600C/601C/611C/602C/612C/603C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

※4 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。
※5 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

品名	品番	価格
●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他		
●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード ※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円
●JIS第1水準漢字ROM ※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・-B	5,800円
●高性能CRTフィルター ※9	BF-68PRO	19,800円
●スキャナ用パラレルボード ※10	CZ-8BN1	27,800円

●品番中の-表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 14/15型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

SHARP

"アート"と呼べる高水準のソフトウェアが

次代のインテリジェンス、ウィンドウ環境をあなたのX68000で。

ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプの美しいデスクトップ(テキスト面/単色4階調+カラー4色、グラフィック面/カラー65,536色中16色)、イベント・ドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。従来のビジュアルシェルとは異なり、今後のアプリケーションソフトが統一された操作環境で実行できるようになります。

SX-WINDOW ver1.0

CZ-259SS　6月発売予定

Communication PRO-68Kのバージョンアップ版です。300BPSから19,200BPSまでの通信速度に対応し、パソコン同士の接続や各種データベースの漢字端末に、またホストコンピュータとのデータ通信に利用できます。さらにMNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポート。その他、高速逆スクロール機能、オートログイン/オートパイロットが可能な自動実行機能、コンカレント機能も装備。行入力機能やスクリーンエディタなど豊富な編集機能も魅力です。また、バイナリファイルを転送するプロトコルとしてXmodem(128/SUM、128/CRC、1K)、Ymodem(G、BATCH、G-BATCH)、TransIt2(TEXT、BINARY)プロトコルもサポートしています。

CZ-257CS　6月発売予定

Communication PRO-68K ver2.0

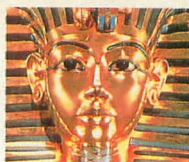
ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Cのソースレベルでデバッグできるソースコードデバッガを搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(約800種)となっています。C言語の最も基本的なK&R仕様を充たし、ANSI規格案に準拠した最新のCコンパイラです。「BASIC-Cコンバータ」、「アセンブラ」、「リンカ」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「ライブラリアン」、「コンバータ」などのツールが装備されています。

CZ-245LS　7月発売予定

C compiler PRO-68K ver2.0

シャープ株式会社　●お問い合わせは…シャープ(株)電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

資料請求券 CZソフト Oh!X

シャープオリジナルソフトウェア
X68000

X68000をサポート。

ビジネスツール

Hyperword
■CZ-251BS　標準価格39,800円(税別)
X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。

CYBERNOTE PRO-68K
■CZ-243BS　標準価格19,800円(税別)
プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。

Stationery PRO-68K
■CZ-240BS　標準価格14,800円(税別)
他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。

TOP給与計算エキスパート
■CZ-228BS　標準価格200,000円(税別)
給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計
■CZ-227BS　標準価格200,000円(税別)
会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K
■CZ-226BS　標準価格29,800円(税別)
自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS　標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS　標準価格9,800円(税別)

DATA PRO-68K
■CZ-220BS　標準価格58,000円(税別)
入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-68K
■CZ-212BS　標準価格68,000円(税別)
スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

アートツール

NEW PrintShop PRO-68K
■CZ-221HS　標準価格19,800円(税別)
オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。

グラフィックライブラリ VOL.1
■CZ-235GS　標準価格8,800円(税別)
暑中見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

グラフィックライブラリ VOL.2
■CZ-236GS　標準価格8,800円(税別)
年賀状を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

サウンドツール

Musicstudio PRO-68K ver.1.1
■CZ-252MS　標準価格28,800円(税別)
24トラック対応MIDIマルチレコーディングソフトMusicstudio PRO-68Kがバージョンアップしました。従来の機能に加え、小節間のコピー及びデリートや、MIDIインプットモニターなど、数々の機能を追加・改良。さらに使いやすくなりました。
※MIDIボード(CZ-6BM1)が必要です。

MUSIC PRO-68K〔MIDI〕
■CZ-247MS　標準価格28,800円(税別)
MIDI対応自動伴奏機能をサポート、簡単な楽譜入力でMIDI演奏が楽しめます。
※MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ〈101曲集〉
■CZ-248MS　標準価格8,800円(税別)
鑑賞用と音楽データ加工作成用からなるライブラリです。

Sampling PRO-68K
■CZ-215MS　標準価格17,800円(税別)
AD PCM機能を活かす高機能サンプリングエディタ。多彩なEDITORを装備、サンプリング音のデータはBASICでも活用できます。

SOUND PRO-68K
■CZ-214MS　標準価格15,800円(税別)
スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

MUSIC PRO-68K
■CZ-213MS　標準価格18,800円(税別)
最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ＆演奏用ツール。

ゲーム

シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI. 1988

シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI. 1989

ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987

ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校
ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988

アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO

アクションゲーム
〈ニュージーランド
ストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989

スポーツゲーム
〈V'BALL〉
■CZ-246AS
標準価格7,900円(税別)
©TECHNOS JAPAN CORP. 1989

バイクレーシングゲーム
〈スーパーハングオン〉
■CZ-238AS
標準価格8,800円(税別)
©SEGA 1987

ジェットヘリ・シミュレーションゲーム
〈サンダーブレード〉
■CZ-239AS
標準価格9,500円(税別)
©SEGA 1987

アクションゲーム
〈ダウンタウン熱血物語〉
■CZ-254AS
標準価格8,800円(税別)
©TECHNOS JAPAN CORP. 1989

開発ツール

OS-9/X68000
■CZ-219SS　標準価格29,800円(税別)
OS-9のもつマルチタスク機能、リアルタイム機能を活かした使い易く機能的なOS環境を提供。これまでのデータ資産も活かせます。
※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0
■CZ-244SS　標準価格9,800円(税別)

THE福袋V2.0
■CZ-224LS　標準価格9,980円(税別)

AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS　標準価格188,000円(税別)

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

SHARP

X68000 information
インフォメーション

EXEクラブが待っている。

●X68000を手にしたら何はともあれ「EXEクラブ」へ。トレンディな仲間が熱烈歓迎。同梱のハガキをすぐに返送してください。折り返しX68000オリジナルの会員証電卓をお手元に……。またEXEクラブニュースや新作ソフト、周辺機器などX68000の最新情報も随時ご送付。各種イベント、フェアへのご招待も。もちろん会費は無料です。
●EXEリーダーズグッズ、プレゼント実施中(EXE会員のみ)あなたの知人、友人やX68000に興味をお持ちの方をぜひご紹介ください。ご紹介くださった方が期間中(平成2年9月30日まで)X68000を購入されますと、あなたにすてきなリーダーズグッズをプレゼント。詳しくはお近くのEXEショップまでお問い合わせください。

NEW X68000、新作ソフト、面白イベント……まるごと見・体・験フェア。

●ご存じ「見体験フェア」、今回のテーマはやはりニューX68000。SUPER-HD/EXPERTII/PROIIの魅力を直にご体験ください。業界注目のSX-WINDOWもまるかじり。MIDIシステムの"トレンドテーマコーナー"、新作ソフト体験コーナー、100インチ液晶プロジェクションによる大迫力のゲームたちに出会えるかも。人気のX68オリジナルグッズも展示即売!! お近くの会場へぜひお立ち寄りください。

NEWミュージックトレンド"MIDI"体験!! 必聴、必見「音遊サウンドライブ」。

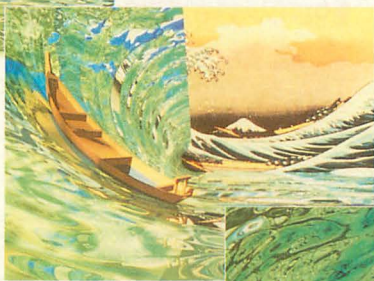
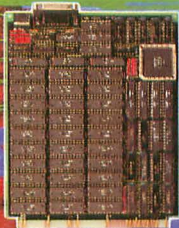
●X68000が創造するパソコントレンド、C.G.の次はMIDI。デジタル世代を自負するならこれに乗り遅れてはいけないのは自明の理。90年代の音楽シーンはMIDIなくしては語れない。楽器のプロもパソコンのプロも、どちらも一度は手を染めてみたいのがこのフィールドだ。そして、音楽さえ好きであれば、どちらも苦手な人も即エンターティナーになれるところがウレシイ。自分の好きな曲を、好きな楽器で、好みのリズムで演奏できる、いま注目度一番のニュートレンド。プロのキーボード奏者による本格的なMIDIライブコンサートで、キミの感性をとがらせよう。今、全国各地で開催中!

パソコンミュージック MIDI X68000 音遊サウンドライブ in Summer

EXE club

彼女が「EXEクラブ」に出かけたら……。

●見体験フェア/音遊サウンドライブ 開催日程

開催月日	開催地区	開催場所	お問い合わせTEL
6/23(土)	神奈川	ICコスモランドあざみ野店フェア	(045)903－1010
6/23(土)・24(日)	東海	ジャスコ豊田店フェア	(0565)35－1761
6/23(土)・24(日)	東京	九十九電機7号店フェア	(03) 253－4199
6/24(日)	東京	T-ZONEフェア	(03) 257－2650
6/30(土)・7/1(日)	中部	コムライン豊田店フェア	(0565)35－1150
6/30(土)・7/1(日)	東海	ジャスコ岡崎店フェア	(0564)23－4960
7/1(日)	神奈川	ウェーブ・アイ大和店オープンフェア	(0466)43－1711(湘南台店)
7/7(土)・8(日)	東北	デンコードー八戸本店 OAフェア	(0178)44－4111
7/14(土)・15(日)	北海道	シャープグランドフェア イン 釧路	(0154)22－9107
7/29(日)	北陸	シャープ見体験フェア イン 富山	(0762)49－1181

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表) シャープ株式会社

アニメーション用ローダー

68用

アニメーションツールボックス「うごくZO」ARGOより発売
価格¥7,800(税込み)C-TRACEクラブ会員価格¥5,800(税込み)
問い合わせTEL.03-5996-4459

98用

フルカラーフレームバッファ「写像」の付属ソフト「anime」が使用できます

C-TRACE TOWNS	¥68,000	
C-TRACE NEWS Ver.3.0(SONY)	¥530,000	新発売
★C-TRACE 98TP	¥610,000	
★C-TRACE 68TP	¥610,000	

表示価格に消費税は含みません。
★の製品は店頭販売いたしておりません。直接当社までお申し込みください。

CasT
株式会社キャスト
●お問い合わせ先●
〒158東京都世田谷区等々力2-1-13
TEL.03-705-1065 FAX.03-705-5224

闇の血族

THE PREDESTINED HOMICIDES #1

美少女名探偵 魅由の繰り広げる
ミステリアスアニメーションアドベンチャー第1弾!!

艶やかなファッション界を襲う奇怪な連続殺人事件。
南米の血に隠された秘密とは?
そして魅由を待ち受ける血族の宿命は?

NOVEL WARE

あたし、魅由。
新宿にあるデザイン・スタジオの、新人A・D(アパレル・デザイナー)。……なんだけどあたしの持ってる妙な「力」みたいなモノ――人の心が判っちゃったり、変にカンが良かったり――のせいで、周りからは「名探偵魅由」なんて呼ばれて、よく相談事を持ち込まれたりしている。で、そんなある日、友達のモデルが、突然、殺されてしまった。
そして、あたしの親友だった唯も……!
これって……ひょっとして連続殺人事件ってヤツ?!

新発売!!

X68000対応 5"-2HD
●ローランド社MT32完全対応
MIDIインターフェイスボードC-2-6BMI
又は、SACOM製SX-68Mが必要です。
(初期のMT-32では、正常に演奏できません。)
標準価格 **8,800円**

伊澤 魅由	姫野 里沙	雪原 リーン
(いざわ みゆ) 19才 誕生日:7月16日 身長:168cm 体重:5?kg	(ひめの りさ) 18才 誕生日:4月2日 身長:163cm 体重:45kg	(ゆきはら リーン) 20才 誕生日:2月10日 身長:170cm 体重:53kg
高校生の時、デザイナーの泉麗子に見込まれ、学生生活を営む傍ら麗子のデザインスタジオ(専門学校)に通い始める。そこで小品の手伝いなどをしながら、デザイナーとして本格的に勉強を開始。2年間の研修期間を終え、高校卒業と同時に麗子の強力な推薦で、現在所属している〈スタジオYo〉に入った。	〈スタジオYo〉の専属モデル。ファッションショー、雑誌モデルを専門としている。平凡な可愛さがウリで、生活の中で"Yo(自己性)"をファッショナブルに演出する――といった〈スタジオYo〉のメイン・コンセプトから考えれば、最もYoらしいモデルと云えるかも知れない。	〈スタジオYo〉の付属学校、「矢萩デザイナーズ・スタジオ」の卒業生。研修期間中「Yoプロデュース」でスタイリスト補助のアルバイトをしていた。現在では、Yoでファッションショーを中心とした若手スタイリストとして活躍中。
趣味:ジョギング、ティー・タイム 好きな食べ物:チーズケーキ 嫌いな食べ物:ブロッコリー 好きな音楽:なんでも大好き、現在はディズニー音楽に夢中とか 家族構成:父、母、妹	趣味:読書、推理ものが好きとかで、本格的シャーロキアンでもある。 好きな食べ物:果物、全部 嫌いな食べ物:ピーマンだけが、苦手 好きな音楽:主に、ニューエイジ・ミュージックをよく聞く。特に好き嫌いなし。 家族構成:父、母、兄	趣味:読書、ジョギング 好きな食べ物:特になし 嫌いな食べ物:コンニャク 好きな音楽:クラッシック 家族構成:父、母、弟

ゲームセンターを賑わした大人気シューティングゲーム
「ジェミニウイング」が、キミのX68Kで今、蘇る!!

ジェミニウイング

Gemini Wing™

幾千の流星が降りそそいだ年、世界は蟲に覆われていた。人々は孤立し、街は滅び、植物に埋め尽くされた。蟲たちはさらに勢いを増し、残された僅かな地さえも蝕んでゆく。そして、ついに最高機密指令第307号、コード名 ジェミニウイングは発動された……!

◆特徴◆
- ●二人同時プレイ可能
- ●5"2HD 2枚組
- ●MIDI対応(※)
  対応楽器 ローランド MT-32 CM-32L CM-64
- ●FM音源、ADPCM対応
- ●ジョイスティック対応
- ●縦横画面モード対応

X68000
全シリーズ対応

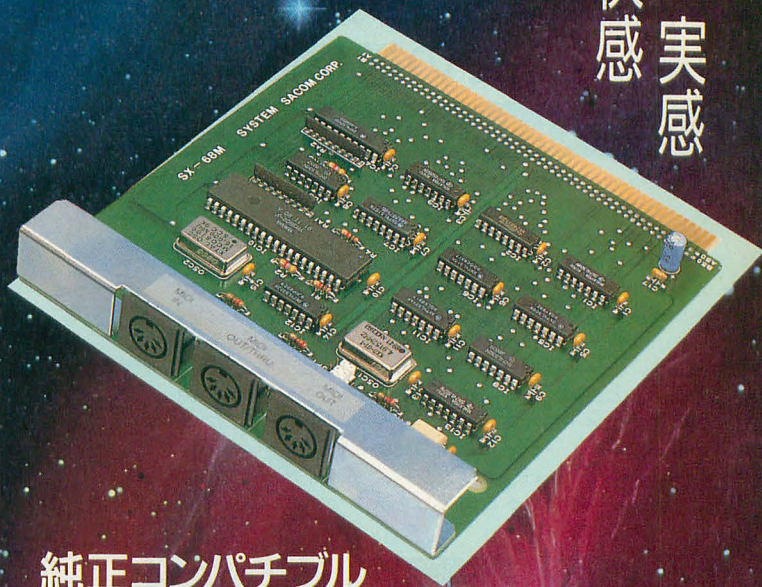
(※)対応機種ごとに、それぞれ違ったBGMをお楽しみいただけます。標準価格8,800円

注)初期のMT-32では正常に演奏できません。

東芝EMIより『38万キロの虚空』CD
7月11日 新発売!! 税抜価格 2,184円 税込価格 2,250円

X68000 SERIES

MIDI INTERFACE BOARD
SX-68M

体感 快感 実感

純正コンパチブル

「SX-68M」は、純正品との互換性を保ちつつ(※)さらに、お求めやすい低価格におさえた、X68000シリーズ専用MIDIインターフェースボードです。
特徴としては、ボード本体に直接MIDI規格のDINコネクタを装着することによって、中間に変換ケーブルを使用する必要がなくなりました。また、クロック部に安定度の高いオシュレーターを採用することにより、さらに信頼度の高いものとなっております。もちろん、従来のMIDIボードをサポートするソフトウェアはそのままお使いになれます。SX-68Mで、あなたもすばらしいMIDIの世界を体験してください。

(※)本ボードは、TAPE SYNC.端子を装備していないため、その機能をサポートするソフトは、ご使用いただけません。また、本ボードは、2枚同時装備ができませんので、ご注意ください。

SX-68M仕様

項目	仕様
品名	MIDIインターフェースボード
規格	MIDI規格 1.0準拠
コントロールLSI	日本楽器(YAMAHA) YM3802
MIDI端子	MIDI OUT 2端子 MIDI IN 1端子
	MIDI OUT 1端子 MIDI THRU 1端子 MIDI IN 1端子
電源	+5V 170mA(本体より供給)
外形寸法	150mm(W)×167mm(D)×23mm(H)
重量	約160g
標準価格	¥19,800

対応ソフト紹介

■38万キロの虚空

■メタルサイト

株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
TEL. 03-635-7609
ソフトウェア部 TEL.03(635)7609
ハードウェア部 TEL.03(635)5145

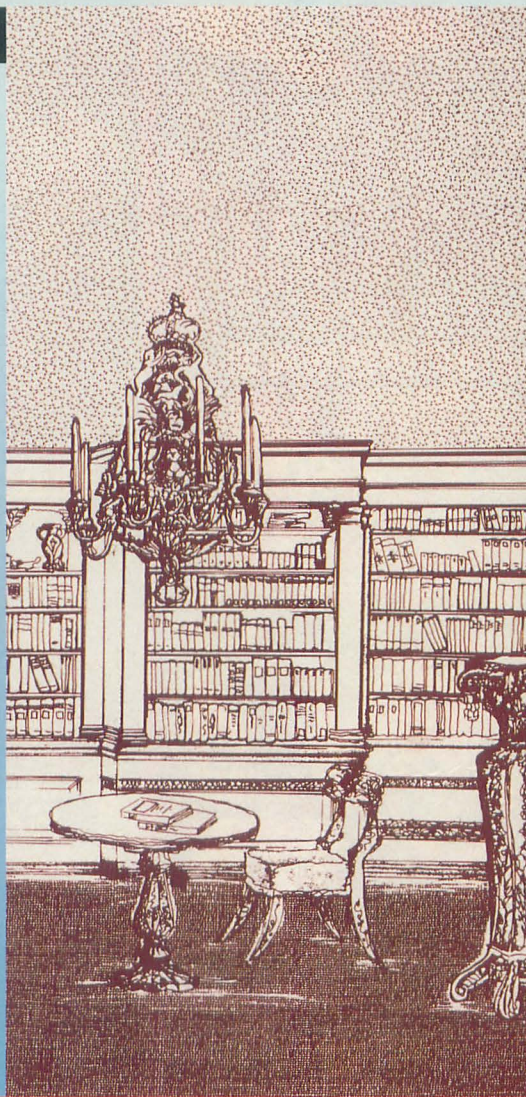
※標準価格には消費税は含まれておりません。

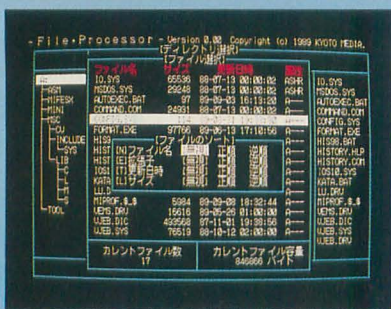
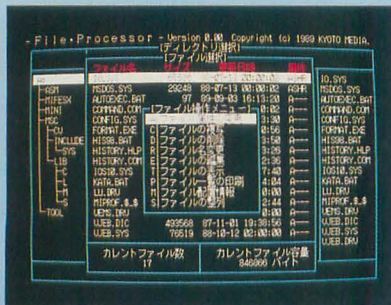
本格的ファイルマネージングソフトウェア

業界の新星、ロゴスシステムがユーザーの希望を1つの形にしました。これは必要だとか便利じゃない、快感だ！

全国有名パソコンショップでお求め下さい。
電話1本での通信販売も受付いたしております。

THE FILE PROFESSORの実力

ディスクのバックアップ、ディスクのエディット、ディスクの初期化、ディスクの比較、ディスクの検査、ディスクの情報、FATのエディット、ファイルの検索、ディレクトリのコピー、ディレクトリの削除、ヴォリュームラベルの設定、ディレクトリの作成、ディレクトリ構造の再読み込み、ディレクトリ構造の印刷、ディレクトリ名の変更、ディレクトリ内容のソート、削除ファイルの復元、ファイル属性の変更、ファイルのコピー/移動、ファイルの削除、ファイルのエディット、ファイルの配置情報、ファイル一覧の印刷、ファイル名の変更、ファイルのソート、ファイル更新日時の変更、ファイルの表示、ファイルの実行、カレンダー、ハードディスクの直接エディット、システム情報の表示、コマンドシェル、現在時刻の変更。

メニュー選択方式を実現!!初心者でも簡単に使える

（画面写真は、98用を開発中のものです。）

ロゴスシステム

このソフトはロゴスシステムのデビュー作です。でも、だからといってなめてもらっちゃぁ困ります。私達は、いろいろなソフトを作りました。そのどれもが他社から発売されていました。出来る事ならば自分達で発売したい／その願いがやっとかないました。

好評発売中！

ロゴスシステム
〒615 京都市右京区西院上今田町17-1 L&Pビル4F
TEL (075) 812-6383　FAX (075) 822-6915

定価28,000円

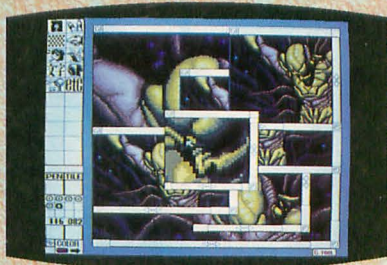
ザインの彩先端。

ニューコンセプトのアートキャンバス「G=ツール」、登場。

X68000ユーザーのクリエイティブマインドに火をつける新感覚のグラフィックツール。これまでのエディタ概念を払拭し、作品に挑むうえで必要不可欠なグラフィックキャラクタ・背景作成のすべてを備えたトータルツールです。ゲームデザインをはじめとしたオリジナルコンピュータアートが驚くほど自由に多彩に描けます。今回はこの「G=ツール」の解析シリーズを展開するにあたって、その驚異の機能概要とシステム環境をご紹介します。

「G=ツール」 新登場。

7月中旬発売予定
¥28,000

「GR EDITモード」と「BG EDITモード」を装備。

グラフィックやスプライトのキャラクタの作成を目的とした「GR EDITモード」、背景やそのキャラクタ作成を目的とした「BG EDITモード」を装備、C.G.を作品として仕上げるうえで必要なエディットツールをすべてサポート。そして、その機能のひとつひとつに新しい感覚が取り込まれています。

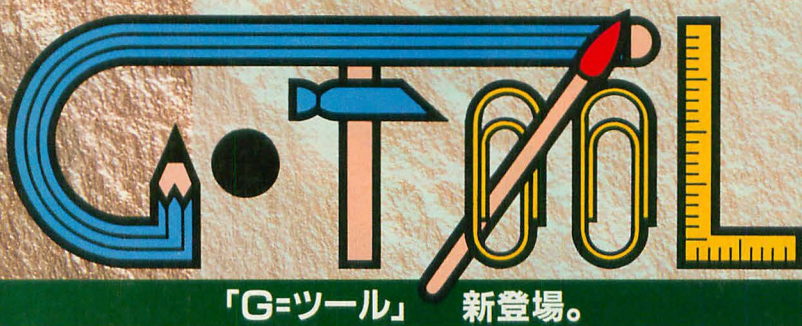
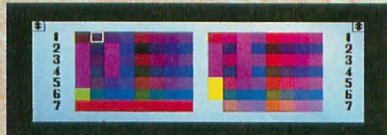
GR EDITモード ●最大12枚の描画ウインドウが開けるマルチウインドウシステム●自分なりのアイコンメニューが作れる便利なユーザーアイコンシステム●マウス機能定義システム●高速メニューウインドウ処理

BG EDITモード ●最高80×80ドットのキャラクタ作成が可能●キャラクタチェックができるアニメーション機能●1～16倍の拡大画面での描画が可能●強力な外部関数を装備●全画面スクロールが可能な背景作成

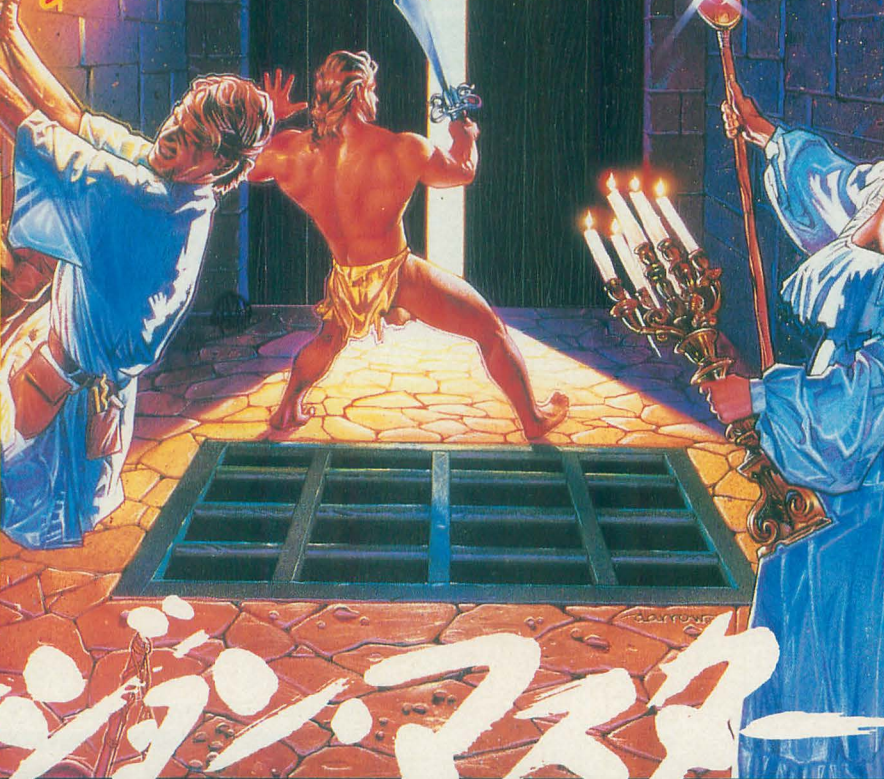
またジョブ画面のメインアイコンには、ファイル/パレット/タイル/ペン/描画動作/編集/文字/マスク/などをわかりやすく配置。それぞれポップアップメニューで仕事が選べます。さらに好きなアイコンを自由に置けるユーザーアイコンエリアを装備、使いやすいシステム環境を提供します。

zainsoft
株式会社ザイン・ソフト
〒676 兵庫県高砂市米田町米田1162-1
TEL.(0794)31-7453

もう逃げられない!

FTL

世界サイズの興奮!

世界中で数々の金字塔を打ち立てたリアルタイムRPG「ダンジョン・マスター」の興奮は本物だった。
3Dグラフィックスに展開される奥の深い迷路、数々のトリック、パーティーを突然襲って来る不気味なモンスター、
組合せと熟練度によって決定される魔法、それぞれの武器によって異なる攻撃方法、
そして何よりもプレイヤーの思考、行動にリアルタイムで反応する見事なゲーム・システム……

これが本当のRPGだ!

■発売 ビクター音楽産業株式会社

通信販売 当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい。(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業㈱(通信販売係)

"夢中宣言'90
by T&E SOFT

'90年後半押し寄せる新たな波。

X68000ビッグ3タイトル連続リリース!!

RUNE WORTH
第1弾 黒衣の貴公子

UNDEAD LINE
第2弾 幻獣鬼

NEW 3D GOLF SIMULATION
第3弾 遥かなるオーガスタ

T&EのBIGウェーブにご期待下さい。

X68000

X68000シリーズ第1弾!

7月13日㈮発売決定

X68000 5"2HD¥8,800 ※表示価格に消費税は含みません。

© 1990 T&E SOFT

●PC-9801VM、UVシリーズ PC-286、386シリーズ、NOTE対応
5"2HD/3.5"2HD 2枚組 ●サウンドボード対応●ジョイスティック対応
※VM,UVはRAM容量640Kバイト以上必要です。
※PC-9801/E/F/M/VF/U2/XAでは、ドライブ、RAM等の増設の如何にかかわらず、作動いたしません。また1ドライブのみ搭載のPC-286L/LE/LFおよびPC-286NOTE Executiveでは、ドライブを増設しても作動いたしません。
※高解像度(640×400ドット)カラーディスプレイをお使いください。液晶ディスプレイにも対応しています。

●PC-8801SRシリーズ・VA、98DO対応 5"2D 5枚組
●サウンドボードII完全対応、ADPCMをフルサポート●ジョイスティック対応(98DOを除く)
●NEC純正128KRAMボード、I/Oデータ機器製RAMボードに対応したキャッシュドライバー搭載

●MSX2/MSX2+(RAM64K以上 VRAM128K以上) 3.5"2DD 3枚組
●MSX-MUSIC(FM音源)対応●ジョイスティック対応

標準価格 各¥8,800
表示価格に消費税は含みません。 MSX is a registered trademark of ASCII Corp.

まだまだ続くBIGウェーブ、お楽しみに!

T&E SOFT
企画・開発・製造・販売 株式会社ティーアンドイーソフト
〒465 名古屋市名東区豊が丘1810番地 PHONE:052-773-7770
●T&Eの最新情報がわかるテレフォンサービス 名古屋(052)776-8500

世にも楽しいシューティングパズル

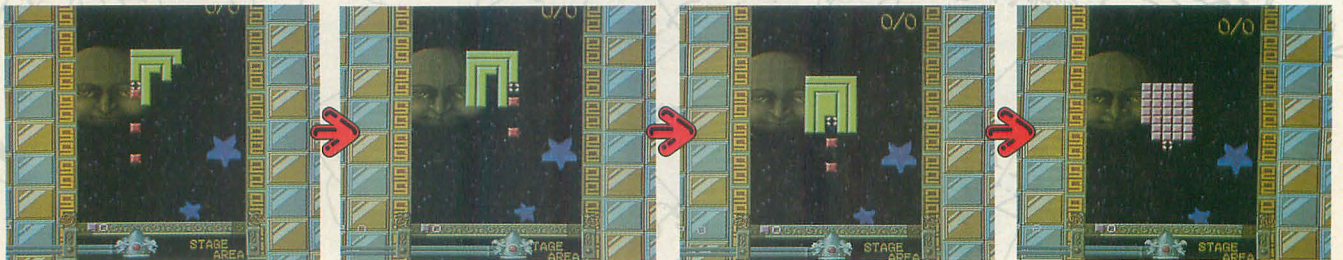
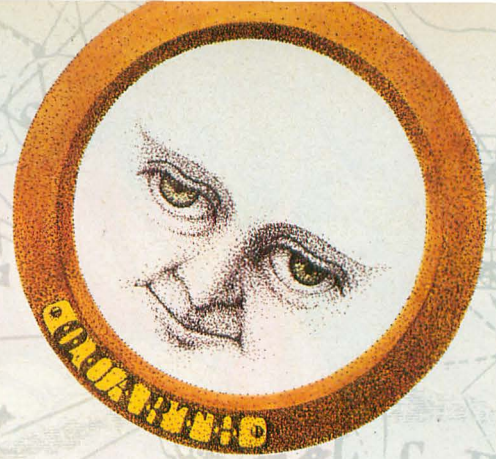
©KONAMI 1990

X68000版
7月6日発売予定
6,800円(税別)

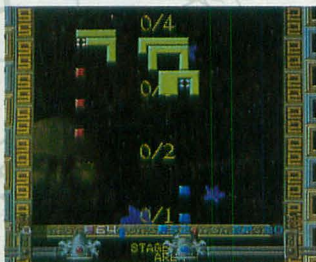
MSX2版 好評発売中 5,800円(税別)

PC-9801版 近日発売予定

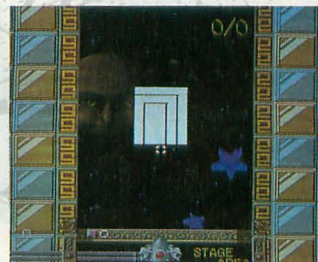
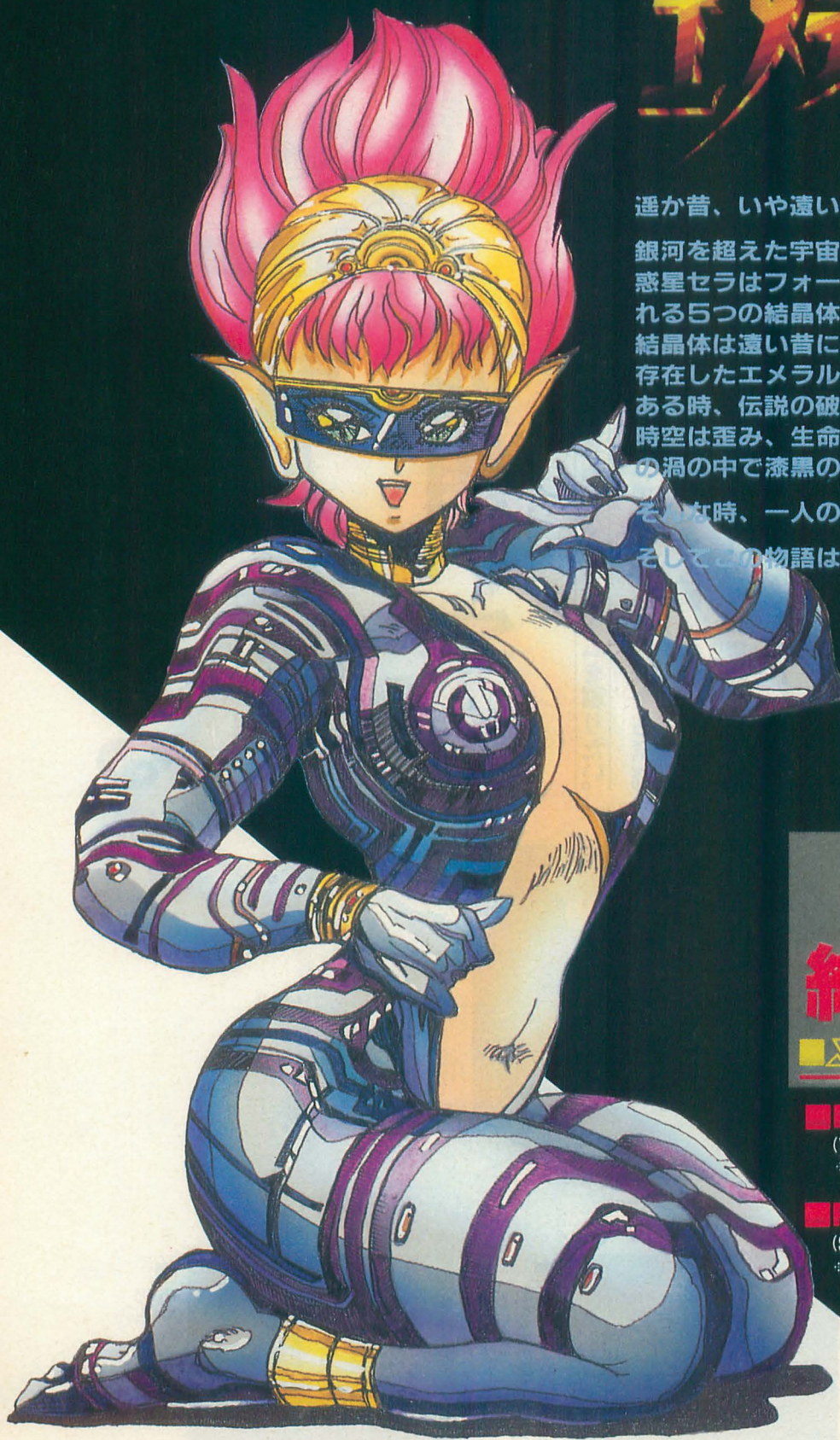
落ちて来るブロックを四角にして消してゆきます。一度にたくさん消すと効率的で得点も大幅アップ。下のラインまで来るとゲームオーバーです。

コナミ株式会社
本社／〒101 東京都千代田区神田神保町3丁目25　大阪支店／〒561 大阪府豊中市庄内栄町4丁目23番18号　札幌営業所／〒060 札幌市中央区北1条西5丁目2番9号　福岡営業所／〒810 福岡市中央区天神2丁目8番30号

歴史に残る前人未踏の四角い宇宙

だれもが夢中になれるゲームを創りたい。

シンプルでいて奥が深い。だれでも気軽に遊べて、いつまでも飽きない。そんなピュアな、ほんとうの意味でのゲームがしたいと思うことがある。ゲームに対する熱い想いをもう一度じっくりと見つめて今、コナミが新たに発進する、楽園ゲームプロジェクト「クォース」。シューティングの楽しさと、パズルの思考性がマッチングした、すでにゲームセンターでは爆発人気の極楽行き超ソフトだ。ほら、もう引力がココロをズルズルと吸いこんでいる。君も、友も、父も、母も、老若男女を巻きこんで、楽園へ行こう。

アイテムブロックが出るとラッキー。

対戦2Pは、敵と相手の両方と戦う恐怖のケンカバトルだ。

協力2Pで、仲直りもできる、ますます熱中の親切設計。

エメラルド伝説
定価各7,800円(税別
遥か昔、いや遠い未来かも知れない。
銀河を超えた宇宙の果てに美しき惑星「セラ」があった。
惑星セラはフォースの力に守られ、それはシーズと呼
れる5つの結晶体から生じていた。
結晶体は遠い昔に滅びたと伝えられる水の惑星「テラ」に
存在したエメラルドという石の中に封印されていた。
ある時、伝説の破壊神「ソドム」が、セラに出現した。
時空は歪み、生命体、物質、そしてシーズまでもが混
の渦の中で漆黒のベールに包まれていった。
そんな時、一人の少女がとある廃虚の地下室で目覚め
そしてこの物語ははじまる………
RPGの世界が新しい広がりをみせる
PC-98/88版
絶賛発売中!!
X68000版7月上旬発売!!
PC9801版(5インチ、3.5インチ2HD)
(VM以降対応、要640K、LT、XL、XAシリーズ、U2を除く)
ご注意……98版ではアナログディスプレイ、16色ボードが必要です。
PC8801版(5インチ2D)
(SR以降対応版、VAシリーズ対応)
ご注意……アナログディスプレイが必要です。
通信販売をご希望の方は…
電話にてお申し込み下さい。
着払いにてお届け致します。
●現金書留の場合………
商品名、機種、メディアを明記の上
右記迄お送り下さい。
宅配便にて急送致します。
(送料弊社負担)
SYSTEM HOUSE OH!
株式会社 システム ハウス オー
〒607 京都市山科区音羽西林町2
☎(075)502-2972

夏ツクモ！ザバーゲン

商品のお申し込みは通販受注センター
フリーダイヤル 0120-377-999
商品のお問い合せは各店、又は通信販売部☎03(251)9911へ！

商品代金2万円以上送料無料!!
※表示価格に消費税は含まれておりません。

今、注目を集めるパソコンミュージック"MIDI"をテーマのライブショーをご案内します。

開催日♪6/23㊏・24㊐午後3回(PM12:30/1:30/2:30) 場所♪九十九電機7号店店頭
キーボードが弾ける人も弾けない人もX68000でMIDIを楽しくMusic！
Musicstudio＆Mu-1の使い方も……
わかり易く解説しますので是非お立ち寄り下さい。

毎年恒例 秋葉原電気まつり
期間：6/21㊍〜7/22㊐
1等 ジャパニーズドリーム(神戸〜横浜の船の旅) ペアで123組ご招待！
2等 JRオレンジカード(10,000円分)3567人へプレゼント！
※¥5,000円以上のお買物でもれなく抽選券をプレゼント！

ナイターセール！
ツクモ7号店、N・C(ニューセンター)店は、この夏夜遅いお客様の為に6/22㊎〜7/28㊏迄の㊎・㊏は
夜8:00迄営業！ごゆっくりお買物ができます！
(N・C店は㊏のみ夜8:00迄営業)
6/20㊌〜8/20㊊迄は、夏ツクモ！ザバーゲンとして5,000円以上のお買い物のお客様にドリンクをプレゼントさせて頂きます。
ツクモの夏は涼しいぞ！

夏ツクモ！ザバーゲン
目玉品 限定5セット 38%OFF
X68000EXPERTセット
CZ-602C ………… ¥356,000
GD-V140 ………… ¥128,000
(14インチDP0.31mm回転台付ノングレア 15〜34kHzオート)
合計定価 ¥484,000
ツクモ特価 ¥298,000 (消費税別途¥8,940)
クレジット例(税込・24回払) 初回¥14,880＋月々¥14,700×23回

NEW
PROII
CZ653C 定価¥285,000
CZ663C 定価¥395,000
●次世代のインテリジェンス、SX-WINDOW搭載●知的ニュースタンダードフォルム●BIOSの改良によりハイスピード処理を実現●3Mバイトの大容量メモリを標準装備●拡張I/Oポート4スロット標準装備

EXPERTII
CZ603C 定価¥338,000
CZ613C 定価¥448,000
●次世代のインテリジェンス、SX-WINDOW搭載●集徴のフォルム、マンハッタンシェイプ●BIOSの改良によりハイスピード処理を実現●3Mバイトの大容量メモリを標準装備

SUPER HD
CZ623C 定価¥498,000
●次世代のインテリジェンス、SX-WINDOW搭載●「チタン」カラーのクォリティブラック●80MBハードディスク搭載●世界標準SCSIインターフェース標準装備●BIOSの改良によりハイスピード処理を実現●3Mバイトの大容量メモリを標準装備

今のハードディスクじゃ容量が足りない！と考えている方へ
ハードディスク シャープより595MBのディスクユニット発売！
光磁気ディスクユニット CZ-6MO1 予約受付中！
SCSIボード CZ-6BS1 予約受付中！
アイテックハードディスク
●IT X640 定価¥158,000 特価¥89,800
●IT X680 定価¥198,000 特価¥118,000

新作ソフト紹介
日本語ワープロ
Hyper Word(CZ-251BS) ……特価¥33,830
ウィンドウシステム
SX-WINDOW(CZ-259SS) ……予約受付中！

X68000用メモリーボード
IOデータ
PIO-6BE1-A 定価¥25,000 特価¥21,500
(ACE&PROシリーズ内蔵用1MB)
PIO-6BE2-2M 定価¥50,000 特価¥42,500
PIO-6BE4-4M 定価¥88,000 特価¥74,500
※2MBと4MBは全てシリーズ対応拡張スロット用

Software tools
(GRAPHIC TOOLS)
マジックパレット……特価¥16,830
Z's STAFF PRO-68K……特価¥49,300
サイクロンExpressα68……特価¥83,300
デジタルクラフト……特価¥33,800
(電子手帳ソフト)
CYBERNOTE PRO-68K……特価¥16,830
Stationery PRO-68K……特価¥12,580
※通信ケーブル CE-300L……特価¥2,520

電子手帳
PA-8600 特価¥24,800
PA-6500 特価¥9,800

プリンター
CZ-8PG1 定価¥130,000 ツクモ特価販売中
CZ-8PG2 定価¥160,000 ツクモ特価販売中
CZ-8PC3 定価¥65,800 特価¥39,800
CZ-8PC4 定価¥99,800 ツクモ特価販売中
IO-735X 定価¥248,000 ツクモ特価販売中
※特価はお電話にてお問い合せ下さい！

通信モデム＆ソフト
アイワ
PV-A24MNP5(モデム) 45% off
定価¥54,800 特価¥29,800
た〜みのる2(ソフト)
ツクモ特価¥15,000

TSUKUMO NET
新規会員募集!!この度、X68000PROのホストシステムへ移行し、3回線までサポートしました。
入会希望の方は7号店荒井まで！
回線番号
☎03(253)2464
ゲストOK！

♪Let's Music♬♪♬♪

MIDIプレイヤーAセット
CM-32L …… ¥69,000
SX-68M …… ¥19,800
Musicstudio Mu-1 …… ¥19,800
合計定価¥108,600
ツクモ特価¥91,800(消費税別途¥2,754)
クレジット例(税込)月々¥5,830×18回払

MIDIプレイヤーBセット
CM-64 …… ¥129,000
SX-68M …… ¥19,800
Musicstudio Mu-1 …… ¥19,800
合計定価¥168,600
ツクモ特価¥144,000(消費税別途¥4,320)
クレジット例(税込)月々¥7,107×24回払

★Musicstudio PRO-68K V1.1又は、MusicPRO68K(MIDI)のソフトの場合には¥8,000プラスになります。

ツクモグローバルカード
国内・外で活躍!!
使って便利、持ってて安心！ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK！しかも18才以上なら学生でもOK！
18才以上なら学生でもOK！
お申し込みは(03)251-9898又は各店で

秋葉原各店
㊫AM10:15〜PM7:00 ㊡毎週木曜日(6/28、7/5を除く)

ツクモは「スーパーX PRO SHOP」です。
PRO STAFF ツクモ
九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

ツクモ7号店 ☎03-253-4199(担当/荒井)
便利で安心な通信販売
通信販売部☎03-251-9911
■ニューセンター店 ☎03-251-0987(担当/福地)
■ツクモ5号店 ☎03-251-0531(担当/川名)
■名古屋1号店 ☎052-263-1655(担当/吉高)
■名古屋2号店 ☎052-251-3399(担当/横山)
■ツクモ札幌 ☎011-241-2299(担当/村井)
★商品のご注文は在庫確認の上お願いします。

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-251-9911へお電話1本！配達日の指定もできます。	月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中！	〒101-91 東京都千代田区神田郵便局私書箱135号 九十九電機株通信販売部 oh!X係	事前に☎でお届け先をご連絡下さい。富士銀行 神田支店(普)No.894047 九十九電機株	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのらせて頂きます。

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

■平成2年 夏のボーナス一括払いOK!!(7月末・8月末)手数料ナシ!!超低金利クレジットをご利用下さい。

チャンス! X68000・SUPER-HD(チタン)=6月末発売!予約受付中!! 送料¥2,000

SX-WINDOW搭載。

●ザ・ワークステーションと呼ぶにふさわしいスーパーな68000.!!新登場.!!
SUPER-HD。

※プレゼント!①MD-2HD10枚 ②アフターバーナー(¥9,200) ③ジョイカード(連射式) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD
●CZ-623C-TN+CZ-613D-TN
定価合計¥633,000…大特価.!!TEL下さい。

※マウス・トラックボール付.!! ディスプレイにはスピーカ2個、チルト台付.!!

12回	?	24回	?	36回	?	48回	?

♡安くてゴメンなさい。今だけヨ.!!

他のディスプレイ① CZ-602D、② 612D、③ CZ-603D、④ CU-21HDの組合せもございますのでお問い合せ下さい。

※超低金利クレジットご利用下さい。1回～60回払い、頭金ナシ!ボーナス1回払い、ボーナス2回払いOK!

オクト限定スペシャルセット

限定スペシャルセット (送料・消費税込み!!)
X68000EXPERT-HD

ラストチャンス!! 早い者勝ち!!

●CZ-612C(BK) (¥466,000)
●CZ-602D(BK) (¥99,800)
●MD-2HD 10枚
●ジョイカード(連射式×2個)
●ゲーム

オクト超特価 **¥364,000**

※ディスプレイ=①CZ-604D ②CZ-605D ③CZ-613D ④CU-21HD との組合せもございます。TEL下さい。

オクト特選 シャープ周辺機器 (送料¥1,000)

●CZ-6BE1 IBM増設RAMボード……(¥35,000)▶特価¥26,500
●CZ-6BE1B 1MB増設RAMボード………(¥28,000)▶特価¥21,000
●CZ-6BE2 2MB増設RAMボード……(¥79,800)▶特価¥60,500
●CZ-6BE4 4MB増設RAMボード……(¥138,000)▶特価¥104,800
●CZ-6BF1 増設用RS-232Cボード……(¥49,800)▶特価¥38,500
●CZ-6BG1 GP-IBボード……………(¥59,800)▶特価¥45,000
●CZ-6BM1 MIDIボード………………(¥26,800)▶特価¥20,500
●CZ-6BN1 スキャナ用パラレルボード‥(¥29,800)▶特価¥22,800
●CZ-6BP1 数値演算プロセッサボード(¥79,800)▶特価¥60,500
●CZ-6BO1 ユニバーサルI/Oボード…(¥39,800)▶特価¥30,500
●CZ-6EB1/BK 拡張I/Oボックス…………(¥88,000)▶特価¥66,800
●CZ-6VT1/BK カラーイメージ・ユニット…(¥69,800)▶特価¥53,000
●CZ-6BL1 LANボード……………(¥268,000)▶大特価
●CZ-8NM2A マウス………………(¥68,800)▶特価¥5,300
●CZ-8NT1 マウストラックボール‥(¥98,800)▶特価¥7,500
●CZ-8NS1 カラーイメージスキャナ……(¥188,000)▶大特価
●CZ-6BC1 FAXボード…………(¥79,800)▶特価¥60,500
●CZ-8TM2 モデムユニット………(¥49,800)▶特価¥38,000
●CZ-64H 増設ハードディスク…(¥120,000)▶大特価
●CZ-6TU GY/BK RGBシステムチューナー…(¥33,100)▶特価¥25,000
●BF-68PRO 高性能CRTフィルター……(¥19,800)▶特価¥15,500
●SX-68M(システムサコム) MIDIボード…………(¥19,800)▶特価¥15,000
●PIO-68BE1-A(I/O DATA) 1MB増設RAMボード……(¥25,000)▶特価¥18,500
●PIO-6BE2-2M(I/O DATA) 2MB増設RAMボード……(¥50,000)▶特価¥37,000
●PIO-6BE4-4M(I/O DATA) 3MB増設RAMボード……(¥88,000)▶特価¥65,000

オクト面白グッズ

アイテック(送料¥1,000)
●IT-X640(¥158,000)…………特価¥103,000
●IT-X680(¥198,000)…………特価¥134,000

モデムコーナー(送料¥1,000)

●MD-1200AIII…………特価¥14,800
●MD-24FS4…………特価¥31,500
●MD-24FS5…………特価¥34,800
●MD-24FP4…………特価¥27,900
●MD-12FS…………特価¥15,000

熱転写カラー漢字プリンター (ケーブル用紙付) 送料¥1,000

CZ-8PC4 ¥99,800 限定
●48ドットサーマルヘッド
●B5～B4まで
●ハガキ可能
●カラー対応
オクト大特価**¥56,800**

① CZ-8PC3(24ドット熱転写カラー漢字プリンター)
定価¥65,800…………特価¥45,000
② CZ-8PK9(24ピン漢字プリンター80桁)
定価¥89,800…………大特価.!!TEL下さい。
③ CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800…………大特価.!!TEL下さい。
④ CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…………大特価.!!TEL下さい。
⑤ CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…………大特価.!!TEL下さい。
⑥ IO-735X(カラーイメージジェット)
定価¥248,000…………大特価.!!TEL下さい。

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価¥16,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!
使い易いデスクです。
1245(H)×614(W)×600(D)
特価¥12,000

X68000ソフト大セール実施中※ゲームソフトオール25%off

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0
(シャフト)定価¥58,000
オクト特価¥40,000

〈データベース〉●KAMIKAZE
(サムシンググッド)定価¥68,000
オクト特価¥46,000

〈グラフィック〉●C-TRACE68
(キャスト)定価¥68,000
オクト特価¥51,000

〈C言語〉●C & Professional Pack
(マイクロウェアジャパン)定価¥58,000
オクト特価¥44,000

〈グラフィック〉●サイクロン エキスプレス
定価¥78,000
オクト特価¥58,000

〈グラフィック〉●デジタルクラフト
定価¥39,800
…………オクト特価¥28,000

〈ワープロ〉●ハイパーワード
定価¥39,800 CZ-251BS
…………オクト特価¥29,800

型名	商品	定価	特価
CZ-211LS	Ccompiler PRO-68K	¥39,800	¥28,800
CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
CZ-213MS	MUSIC PRO68K	¥18,800	¥13,500
CZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
CZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
CZ-219SS	OS-9/X68000	¥29,800	¥21,000
CZ-220BS	DATA PRO-68K	¥58,000	¥41,000
CZ-221HS	New Print Shop PRO-68K	¥19,800	¥14,300
CZ-223CS	Communication PRO-68K	¥19,800	¥14,300
CZ-224LS	THE 福袋 V2.0	¥9,900	¥7,500
CZ-226BS	CARD PRO-68K	¥29,800	¥21,300
CZ-241BS	システム手帳リフィル集	¥9,800	¥7,500
CZ-242BS	活用フォーム集	¥9,800	¥7,500
CZ-244SS	Homan 68K Ver.2.0	¥9,800	¥7,500
CZ-247MS	MUSIC PRO-68K〔MIDI〕	¥28,800	¥20,800
CZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
EW		¥38,000	¥29,800
G-68K		¥14,800	¥11,400
E-68K		¥19,800	¥15,300

店頭ゲームソフトオール25%off!ビジネスソフト25%より特価中

●尚、送料として1ケ¥500、2ケ¥700、3ケ以上で¥1,000となります。(税別)

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2%	3回	2.5%	6回	3.5%	10回	5%
12回	5%	15回	7.5%	18回	9%	20回	10%
24回	11%	30回	14.5%	36回	15.5%	48回	20%

振込先
富士銀行 久ヶ原支店 当No.1824
三菱銀行 蒲田支店 当No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※連休のお知らせ=7/31(水)、8/1(水)は連休です。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

超低金利クレジットOK

注目!! 夏のボーナス一括払い 手数料(金利)無料

(7月末・8月末どちらかご指定下さい)

またまた 秋葉原で おなじみの

6/15〜7/15

●お近くの方はお
●本体単品で特
●ビジネスソフト定

モデム(AIWA) 50台限定 (送料¥1,000)
PV-A24MNP5 (定価¥54,800)
● MNPクラス5
● 2400bps
限定特価**¥26,500**
(送料・消費税込¥28,325)

CYBER STICK
● CZ-8NJ2
(定価¥23,800)
超特価!!
▶価格はTEL下さい

X68000シリーズ専用 特価¥16,480
MIDIインターフェースボード
SX-68M(サコム)
(純生コンパチ)定価¥19,800
送料・消費税込み!

X-1ターボZIII 特別ご提供品!! 台数限定
● CZ-888C+CZ-860D+M-2HD(10枚)
定価¥269,600▶特価**¥164,800**
・ジョイカード
・ゲーム3種
・パソコンラックⒶ3段
プレゼント中
送料消費税込み!!
(ボーナス併用も有りますTEL下さい)

12回	14,400	24回	7,600	36回	5,300	48回	4,100	60回	3,400

ジョイスティック 送料¥500
● X-1PRO
定価¥9,500▶特価¥7,800
● ASCII STICK
定価¥6,800▶特価¥5,500

NEW X68000EXPERT II/II-HD & PROII/PROII-HD & SUPER-HD (送料・消費税込)

EXPERTII

セットでお買い上げの方に、
● ディスケット10枚
● ゲーム3種
● ジョイカード2ケ
プレゼント中!!

EXPERT II

Ⓐセット:CZ-603C+CZ-604D……定価¥432,800▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-603C+CZ-605D……定価¥453,000▶特価(価格はお電話下さい。)

12回	30,200	24回	15,900	36回	11,000	48回	8,500	60回	7,100

Ⓒセット:CZ-603C+CZ-613D……定価¥473,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓓセット:CZ-603C+CU-21HD……定価¥486,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

EXPERTII-HD

セットでお買い上げの方に、
● ディスケット10枚
● ゲーム3種
● ジョイカード2ケ
プレゼント中!!

EXPERT II-HD

Ⓐセット:CZ-613C+CZ-604D……定価¥542,800▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-613C+CZ-605D……定価¥563,000▶特価(価格はお電話下さい。)

12回	37,700	24回	19,800	36回	13,700	48回	10,600	60回	8,900

Ⓒセット:CZ-613C+CZ-613D……定価¥583,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓓセット:CZ-613C+CU-21HD……定価¥596,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

PROII

セットでお買い上げの方に、
● ディスケット10枚
● ゲーム3種
● ジョイカード2ケ
プレゼント中!!

PRO II

Ⓐセット:CZ-653C+CZ-604D……定価¥379,800▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-653C+CZ-605D……定価¥400,000▶特価(価格はお電話下さい。)

12回	26,800	24回	14,100	36回	9,700	48回	7,600	60回	6,300

Ⓒセット:CZ-653C+CZ-613D……定価¥420,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓓセット:CZ-653C+CU-21HD……定価¥433,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

PROII-HD

セットでお買い上げの方に、
● ディスケット10枚
● ゲーム3種
● ジョイカード2ケ
プレゼント中!!

PRO II-HD

Ⓐセット:CZ-663C+CZ-604D……定価¥489,800▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-663C+CZ-605D……定価¥510,000▶特価(価格はお電話下さい。)

12回	34,100	24回	17,900	36回	12,400	48回	9,600	60回	8,100

Ⓒセット:CZ-663C+CZ-613D……定価¥530,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓓセット:CZ-663C+CU-21HD……定価¥543,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

SUPER-HD

セットでお買い上げの方に、
● ディスケット10枚
● ゲーム3種
● ジョイカード2ケ
プレゼント中!!

SUPER-HD

Ⓐセット:CZ-623TN+CZ-604D……定価¥592,800▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-623TN+CZ-605D……定価¥613,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓒセット:CZ-623TN+CZ-613D……定価¥633,000▶特価(価格はお電話下さい。)

12回	42,700	24回	22,500	36回	15,500	48回	12,100	60回	10,100

Ⓓセット:CZ-623TN+CU-21HD……定価¥646,000▶特価(価格はお電話下さい。)

12回	?	24回	?	36回	?	48回	?	60回	?

X68000シリーズ 〜P&Aスペシャルセット=限定誌上販売!!

台数限定 送料、消費税込み

セットでお買上げの方に、
● ディスケット10枚 ● ゲーム3種 ● ジョイカード2個 プレゼント中

EXPERT

● CZ-602C+CZ-612D……定価¥475,800▶特価**¥306,000**
● CZ-602C+CZ-604D……定価¥450,800▶特価**¥300,000**
● CZ-602C+CZ-605D……定価¥471,000▶特価**¥320,000**
● CZ-602C+CZ-613D……定価¥491,000▶特価**¥336,000**
● CZ-602C+CU-21HD……定価¥504,000▶特価**¥338,000**

EXPERT-HD

● CZ-612C+CZ-612D……定価¥585,800▶特価**¥375,000**
● CZ-612C+CZ-604D……定価¥560,800▶特価**¥369,000**
● CZ-612C+CZ-605D……定価¥581,000▶特価**¥386,000**
● CZ-612C+CZ-613D……定価¥601,000▶特価**¥403,000**
● CZ-612C+CU-21HD……定価¥614,000▶特価**¥407,000**

PRO-HD

● CZ-662C+CZ-612D……定価¥527,800▶特価**¥339,000**
● CZ-662C+CZ-604D……定価¥502,800▶特価**¥333,000**
● CZ-662C+CZ-605D……定価¥523,000▶特価**¥352,000**
● CZ-662C+CZ-613D……定価¥543,000▶特価**¥368,000**
● CZ-662C+CU-21HD……定価¥556,000▶特価**¥372,000**

P&A超低金利クレジットをご利用ください!!

● 本広告の掲載の商品の価格については、消費税は含まれておりません。 ● 営業時間=平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

回～84回払いまでOK‼

★頭金なし！★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕‼

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK！ TELください。

全国通販

超特価でクレジットが組める‼

X68000用ソフトコーナー(送料1ヶ～5ヶまで¥500)

商品	定価	特価
Z's STAFF PRO68K Ver2.0(ツァイト)	¥58,000	特価¥39,700
Z's TRIPHONY デジタルクラフト(ツァイト)	¥39,800	特価¥29,300
テラッツォ(ハミングバード)	¥19,800	特価¥15,800
KAMIKAZE(サムシング・グッド)	¥68,800	特価¥46,000
EW&EI(イースト)	¥38,800	特価¥28,800
C&Professional Pack(マイクロウェアジャパン)	¥58,800	特価¥43,000
Final Ver3.2(エーエスピー)	¥38,000	特価¥30,000
DATA PRO68K CZ220BS	¥58,000	P&A特価
CARD PRO68K CZ226BS	¥29,800	TEL下さい！
C compiler PRO68K CZ211LS	¥39,800	特価¥32,000
OS-9/X68000 CZ219SS	¥29,800	P&A特価 TEL下さい。
AI-68K CZ234LS	¥188,000	特価¥143,000
THE 福袋V2.0 CZ224LS	¥9,900	特価¥7,700
SOUND PRO68K	¥15,800	特価¥12,500
MUSIC PRO68K CZ213MS	¥15,800	P&A特価 TEL下さい。
Sampling PRO68K CZ215MS	¥17,800	特価¥14,000
MUSIC-studio PRO68K 237MS	¥15,800	P&A特価 TEL下さい。
MUSIC-PRO68K〔MIDI〕247MS	¥28,800	特価¥22,000
New-print Shop 221HS	¥19,800	P&A特価
Communication 223CS	¥19,800	TEL下さい。
C-TRACE68 Ver.3.0(キャスト)	¥98,000	特価¥77,000
サイクロンEXPRESS α68	¥98,000	特価¥72,000
G68K Ver2 PRO	¥22,000	特価¥16,300
THE FILE PROFESSOR(ロゴシステム)	¥28,000	特価¥20,500
Gツール(ザインソフト)	¥28,000	特価¥20,500
たーみのる2(SPS)	¥17,800	特価¥13,500
マジックパレット(ミュージカルプラン)	¥19,800	特価¥14,900
Hyper word CZ-251BS	¥39,800	特価¥30,900

●ゲームソフト20%OFF OK‼(一部ソフト除く)

周辺機器コーナー(送料¥1,000)

	商品	定価	特価
Ⓐ	CZ-8NSI	¥188,000	特価¥145,000
Ⓑ	CZ-6VTI	¥69,800	特価¥54,000
Ⓒ	CZ-6TU	¥33,100	特価¥25,000
Ⓓ	BF-68PRO	¥19,800	特価¥15,500
Ⓔ	CZ-6BEI	¥35,000	特価¥26,500
Ⓕ	CZ-6BEIA	¥38,000	特価¥28,600
Ⓖ	CZ-6BE2	¥79,800	特価¥60,000
Ⓗ	CZ-6BE4	¥138,000	特価¥107,000
Ⓘ	CZ-6BFI	¥49,800	特価¥38,200
Ⓙ	CZ-6BPI	¥79,800	特価¥61,000
Ⓚ	CZ-6BMI	¥26,800	特価¥20,300
Ⓛ	CZ-6EBI	¥88,000	特価¥67,500
Ⓜ	AN-S100	¥36,600	特価¥28,500
Ⓝ	CZ-6SDI	¥44,800	特価¥35,000
Ⓞ	CZ-8PC3	¥65,800	P&A超特価 TEL下さい。
Ⓟ	CZ-8PC4	¥99,800	P&A超特価 TEL下さい。
Ⓠ	CZ-8PG1	¥130,000	P&A超特価 TEL下さい。
Ⓡ	CZ-8PG2	¥160,000	P&A超特価 TEL下さい。
Ⓢ	CZ-8PK10	¥97,800	P&A超特価 TEL下さい。
Ⓣ	CZ-6PVI	¥198,000	特価¥153,000
Ⓤ	IO-735X	¥248,000	特価¥190,000
Ⓥ	CZ-8BSI	¥23,800	特価¥19,000
Ⓦ	PIO-6BE1-A(I/O DATA)	¥25,000	特価¥18,200
Ⓧ	PIO-6BE2-2M(I/O DATA)	¥50,000	特価¥36,800
Ⓨ	PIO-6BE4-4M(I/O DATA)	¥88,000	特価¥64,800

中古パソコンはP&Aにおまかせ‼

その場で高価現金買取り・高価下取りOK‼

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……………価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……………現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々¥1,000円からOK‼ ●ボーナス払いOK(夏冬10回までOK)
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK‼

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日＝第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

X68000用ハードディスク(送料¥1,000)

アイテム
- ●HXD-040(40MB/23ms)…………定価¥118,000▶特価¥88,000
- ●HXD-042(増設用)…………定価¥128,000▶特価¥95,000

アイテック
- ●ITX-640(40MB/28ms)…………定価¥158,000▶特価¥101,000
- ●ITX-680(80MB/20ms)…………定価¥198,000▶特価¥131,000

プリンター(ケーブル・用紙付)限定5台 新品(送料¥1,000)

- ●CZ-8PC3(カラー漢字24ドット熱転写プリンター) 定価¥65,800……特価¥39,800
- ●CZ-8PK8(24ピン漢字プリンター136桁) 定価¥152,000……特価¥75,800
- ●CZ-8PC4 P&A特選‼(カラー漢字48ドット熱転写プリンター) 定価¥99,800……特価¥59,000

モデムコーナー (送料¥1,000)

	商品	定価	特価
Ⓐ	MD-24FS5(オムロン)	¥49,800	特価¥34,800
Ⓑ	MD-24FS7(オムロン)	¥64,800	特価¥45,000
Ⓒ	コムスター2424/4(NEC)	¥38,800	特価¥28,000
Ⓓ	コムスター2424/5(NEC)	¥44,800	特価¥32,000

P&A特選パソコンラック (送料無料) 移動自由(キャスター付)

中古パソコン 送料¥2,000

商品	価格
X-68000セット	¥210,000
X-68000ACEセット	¥240,000
X1ターボZセット	¥100,000
X-1G/30セット	¥39,000
CZ-822C	¥15,000
CZ-830C	¥25,000
CZ-856C	¥45,000
CZ-870C	¥55,000
CZ-881C	¥65,000
CZ-820D	¥10,000
CU-14GB	¥5,000
CU-14BD	¥25,000
CU-14AG2	¥30,000
CU-14H2	¥30,000
CZ-8PC2	¥25,000
CZ-8PK6	¥32,000

通信販売お申し込みのご案内

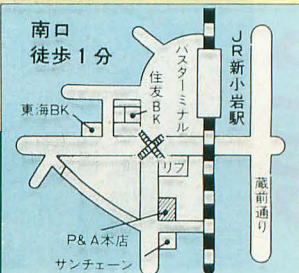
〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	2.5	3.5	5.0	5.0	9.0	10.5	14.5	19.0	24.5	32.0	38.5

●マイコン
●ビデオ
●ビデオテープ

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148(代) FAX.03-651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

スケールが違う!

THE COMPUTER館

日本一のコンピュータ館です

より専門化されたビジネスシステムたち

4F

3F

●4F ビジネスシステムフロアは、様々な業種や業務別に専門化されたシステムがラインナップされています。例えば、卸・小売業に役立つ、店舗POSシステムや、製造業で使われる、部品や在庫の管理システム。さらには、教育関連の成績処理や翻訳、法律情報システムと言ったぐあいにこの他にもあらゆる業種に対応するシステムが体験できます。また業務別では、会社組織の様々な部署で役立つ、ソフトを含めたシステムが体験できるようになっています。

パソコンソフト・パソコン書箱30,000点展示

5F

5F

●建築・機械設計のお仕事に携わっている方にお奨めしたいのが、正確で美しい図面を素早く作製できるCADシステムです。さらに、急な図面の変更にも対応し、高性能プロッタで美しく作製しなおせます。また、このフロアには、コンピュータグラフィックスやCAMと言って、小さな彫刻物のサンプルを製作する優れたシステムもございます。

ザ・コンピュータ館フロアごあんない

6F イベントフロア
催事場

5F CAD&グラフィックス フロア
CAD・CAM・グラフィックスのソフトから周辺機器及びシステムまで/ワークステーション/LaOX NET/パソコン通信用機器及び体験コーナー

4F ビジネスシステム フロア
業種別パソコンコーナー(小売業、不動産業、製造業、教育、広告、飲食店など業種にあわせたソフト及びシステムを動かし、実際に体験できるコーナーです) 業務別パソコンコーナー(社長室、営業部、総務部、経理部、OA推進部など業務にあわせたソフト及びシステムを動かし実際に体験できるコーナーです) ラップトップパソコン 周辺機器 帳票 セミナールーム

3F ビジネスソフト フロア
ビジネスソフト/帳票
IBM/J-3100/AX/Macintosh

2F アクセサリー&ワープロ フロア
ワープロ/ワープロ周辺機器/電子手帳/OAシステムインテリア/コピー/ファクシミリ/アクセサリー/サプライ/セミナールーム

1F ホビー&BOOK フロア
パソコン関連書籍・雑誌/ホビーパソコン・周辺機器/ゲーム・ホビーソフト/プレイゾーン

●秋葉原電気まつり「THE AKIHABARA EXCITING FESTA」開催。期間/6月21日(木)〜7月22日(日) ※抽選で「ジャパニーズ・ロマンチック・クルーズ」をペアでご招待。

情報鮮度が違う！

日本のガリバー「PC-9801」展

期間/6月15日㊎〜7月1日㊐・**会場**/6Fイベントフロア

●豊富なソフトウェアに、豊富な周辺機器で数多くのユーザーに愛されるPC-9801シリーズ。その数々の98の様々な面に出会える楽しさいっぱいの17日間です。98ユーザーの方も、これから購入の方も、目が離せない「PC-9801」展です。

❶PC-9801オールマシン展

現在活躍中のPC-9801シリーズのフルラインナップを始めとして、モニター・ハードディスク・プリンタなどの周辺機器を集中展示いたします。98本体はもちろんのこと、周辺機器選びでお困りの方にはうってつけの催しです。

❷PC-9801ペリフェラル展（周辺機器サードパーティー大集合）

PC-9801シリーズと、その関連ソフトを100%活用するための周辺機器が大集合。プリンタ、ハードディスク、各種ボード、キーボード、マウスなど、あなたのシステム環境をさらによりよくするためにお役立てください。

❸PC-9801 スーパーセミナー ■期間/6月15日㊎〜21日㊍

㋐MS-DOS入門セミナー（6/15㊎〜17日㊐） ①13:30〜14:30 ②15:00〜16:00 ③16:30〜17:30

●MS-DOSとは何か？から始まり、MS-DOSの歴史まで、MS-DOSの基礎知識を身につけていただくためのセミナーです。

㋑PC-9801入門セミナー（6/18㊊・19㊋） ①13:30〜14:30 ②15:00〜16:00 ③16:30〜17:30

●何げなく使われている、bit・RAM・ROM・クロックなどの専門用語を始めとした、98をより納得してご利用していただくためのセミナーです。

㋒周辺機器セミナー（6/20㊌・21㊍）
20日㊌ ①13:30〜14:30（ハードディスク） ②15:00〜16:00（EMSボード） ③16:30〜17:30（ハードディスク）
21日㊍ ①13:30〜14:30（EMSボード） ②15:00〜16:00（ハードディスク） ③16:30〜17:30（EMSボード）

●PC-9801シリーズに使われる、ハードディスクとEMSボードの基礎知識から、応用に至るまでをわかり易く解説するセミナーです。

※スーパーセミナーのお申し込み、お問い合わせは、当店係員にお申し付けください。多数のお申し込みが予想されますのでお早めに。

❹98アプリケーション活用クリニック

■**期間/6月22日㊎〜24日㊐**

PC-9801シリーズ対応の著名ソフトをうまく活用しきれない、又は活用方法がわからないなどの理由で、お困りの方！直接ご自身で ソフトハウス に、自分の使用方法についてのアドバイスが得られるチャンスです。尚、ソフトハウスは、限りられておりますので、詳細につきましては当店係員にお尋ね下さい。

❺ゲーム大会

■**期間/6月29日㊎〜7月1日㊐**

PC-9801シリーズ対応の、今、話題のゲームソフトで、エキサイティングな大会を行います。奮ってご参加ください。

※日本のガリバー「PC-9801」展の詳細につきましては、当店6Fイベント係りに、お尋ね下さい。

パソコンソフトセミナーのご案内

月　日	時　間	セミナー名	ソフトハウス	ジャンル	会場
6月18日㊊	17:00〜19:00	商　魂	PCA	販売管理	4F
6月19日㊋	17:00〜19:00	アシストカード（PC9801用）	アシスト	データベース	4F
6月20日㊌	17:00〜19:00	KOAテクノメイト	高電社	ワープロ	4F
6月21日㊍	17:00〜19:00	Dyna CAD＋Dyna PERS3	ダイナウェア	CAD	4F
6月24日㊐	14:00〜16:00	ロゴライター	DTP	ロ　ゴ	4F
6月25日㊊	17:00〜19:00	初めてさわるTOP財務会計	オービック	会　計	4F
6月26日㊋	17:00〜19:00	UP2システム	ダイナウェア	総合型	4F
6月27日㊌	17:00〜19:00	MS-WINDOWS体験セミナー	日本ソフトバンク	ウインドウズ	4F
6月28日㊍	17:00〜19:00	Aldus Page Maker	日本ソフトバンク	DTP	4F
6月29日㊎	17:00〜19:00	マイクロソフトエクセル	日本ソフトバンク	表計算	4F
7月1日㊐	14:00〜16:00	毛筆わーぷろ（入門）	富士ソフトウェア	ワープロ	4F
7月2日㊊	17:00〜19:00	ロータス1・2・3紹介コース	ロータス	表計算	4F
7月3日㊋	17:00〜19:00	P1.EXE ワープロ入門	dBソフト	ワープロ	4F
7月5日㊍	17:00〜19:00	初めてさわる一太郎	ジャストシステム	ワープロ	4F
7月6日㊎	17:00〜19:00	PCA会計II	PCA	会　計	4F
7月7日㊏	14:00〜16:00	データベース入門 THE CARD3	アスキー	データベース	4F
7月8日㊐	14:00〜16:00	EXCEL（初級編）	マイクロソフト	表計算	4F
7月9日㊊	13:00〜15:30	ハードディスク入門	アイテック	ハード	4F
7月9日㊊	16:00〜18:30	ハードディスク入門	アイテック	ハード	4F
7月10日㊋	15:00〜17:00	How to 社内報 UPシリーズ	ダイナウェア	DTP	4F

※都合により、変更する場合がありますがその際はご了承ください。

ワープロセミナーのご案内

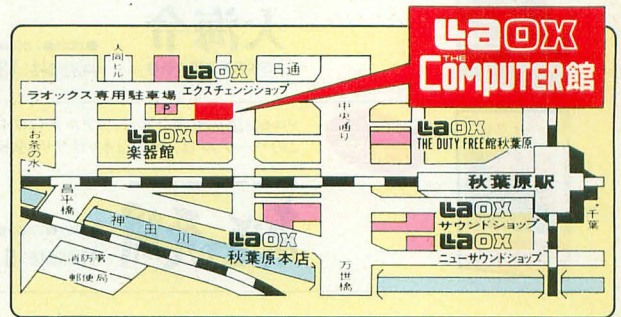
月　日	時　間	セミナー名	機　種	メーカー	会場
6月19日㊋	17:00〜19:00	電子手帳セミナー	DK5000	カシオ	2F
6月21日㊍	14:00〜16:00	RUPOセミナー中級	JW95G	東　芝	2F
6月25日㊊	16:00〜18:00	キヤノワードセミナー	ALPHA50	キヤノン	2F
6月28日㊍	17:00〜19:00	OASYSセミナー中級	OA30AXS	富士通	2F
7月4日㊌	17:00〜19:00	シャープ新製品セミナー	WDA340	シャープ	2F
7月5日㊍	17:00〜19:00	OASYSセミナー初級	OA30AXS	富士通	2F

※都合により、変更する場合がございますがその際はご了承ください。

受講ご希望の方は

●各セミナーとも専門の係員が、実際に皆さんと共にパソコンやワープロを起動させながら説明していきます。ぜひ、この機会をご利用ください。尚、各セミナーとも定員になり次第しめ切らさせていただきますのでご了承ください。お申し込み、お問い合わせは、お電話にて当店セミナー係員にお申し付けください。多数のお申し込みが予想されますので、セミナー受講のご予約はお早めに願います。

その他、楽しく役立つイベントやソフトのデモ実演など最先端の情報がいっぱいです。

さらに、3,500名様にJRギフトカード1万円分をプレゼント！詳しくは当店係員まで。

ザ・コンピュータ館

TEL/03-5256-3111

〒101 東京都千代田区外神田1-7-6（秋葉原駅徒歩4分）Ⓟ駐車場完備

ARTDINK

■アートディンクからビッグニュース!
A賞B賞あわせて510名様に豪華プレゼント!
1990年9月12日まで

A賞	次のうちいずれかひとつを、抽選で合計10名様にプレゼント。 NEC 98NOTE 3名様 緑電子 40MBハードディスク 3名様 シャープ 20MBハードディスク 1名様 エプソン 熱転写プリンタ 2名様 シャープ 熱転写プリンタ 1名様
B賞	オリジナルハードケース入フロッピーディスク(5枚セット)を抽選で毎月100名様、合計500名様にプレゼント。
応募方法	パッケージに同封されたユーザ登録ハガキに必要事項をご記入のうえ当社宛ご返送ください。
応募締切	1990年9月12日まで(当日消印有効)
発表・発送	A賞の当選発表は、当選者本人への通知、及び10月発売のパソコン専門誌(「Oh!PC」「Login」等)の誌上広告にて。賞品発送は10月中旬です。B賞の当選発表は、賞品の発送(毎月10日締切15日発送)をもって、かえさせていただきます。

鉄道王の夢を追うオムニバスタイプ・シミュレーションゲーム。
マップ数は、なんと10! ほぼ世界中を手中にできる。

本格的鉄道シミュレーションゲーム
A列車で行こうII

●X68000●5"2HD(4枚組)
特別価格:12,800円

30年のプロゴルファー人生を歩むロールプレイング付。
他に類をみない2WAYゴルフシミュレーション。

●X68000●5"2HD(2枚組)
標準価格:9,500円

太平洋戦争の激闘をモチーフにしたウォーシミュレーション。
6つのシナリオを収め、FD8枚組の大作バージョンで登場。

SUPER REAL TIME WAR SIMULATION GAME

●X68000●5"2HD(8枚組)
特別価格:13,800円

光の刃、陽子砲が星海に躍る。そして敵機を切り裂く。
太陽系の運命を賭けて戦うスペースウォーシミュレーション。

●X68000●5"2HD(3枚組)
標準価格:9,500円

ソロモン諸島近海を舞台にリアルタイムの局地戦を展開。
このバージョンは、合計12本のシナリオを収録した豪華版。

SUPER REAL TIME WAR SIMULATION GAME

●X68000●5"2HD(4枚組)
特別価格:9,500円

戦車72両、2,000を越す将兵を率いて、
無敵のドイツ機甲師団に挑む。

●今秋発売予定

表示価格に消費税は含みません。

X68000ユーザの皆様へ

すでに、各報道でご承知のことと存じますが、4月13日に発売した初期出荷バージョン**「ファーサイドムーンX68000版」(機種表示部分が真赤なもの)**の一部にコンピュータウィルスが発見され、X68000ユーザの皆様のみならず、社会全体をおさわがせしましたことに深くお詫び申し上げます。

問題となっているウィルスは、X68000本体のS-RAMに付着する「NX68K IPL V1.02」という種類のもので、症状は今年の7月になったらドライブに差し込まれたディスクのデータを破壊するかも知れないという極めて悪質なものです。

しかし、このウィルスを撲滅することができる「ワクチン」がすでに準備されており、ただいま、弊社登録ユーザの皆様に無料で配布いたしております。
まだユーザ登録をされていないお客様は、大至急、当社まで「ユーザ登録ハガキ」をご返送ください。

未来あるパソコンライフ、安心してコンピュータに親しむことができる社会環境のためにも、ユーザの皆様方の絶大なるご支援をお願いする次第です。

何卒、ご理解ご協力をたまわりますよう、よろしくお願い申し上げます。

株式会社 アートディンク 〒275 千葉県習志野市津田沼2-11-20
TEL 0474-77-7541(ユーザサポート専用)

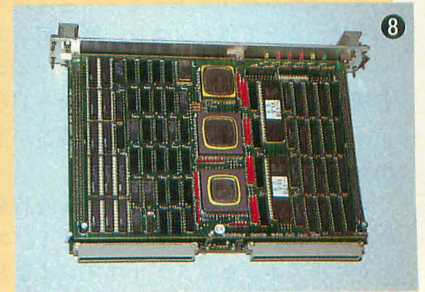
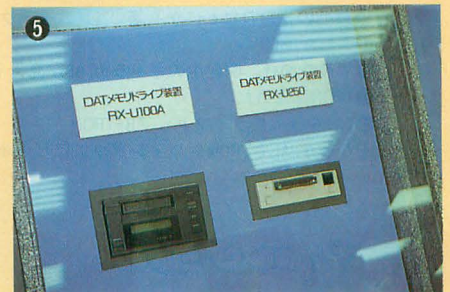
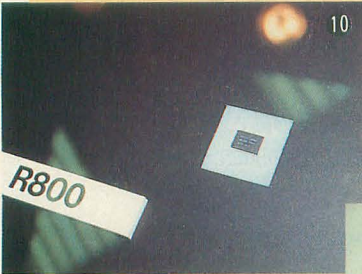
●お買い求めは、全国ショップにて。
●通信販売(送料無料)、住所・氏名・電話番号・商品名・機種名・メディア名を明記して消費税3%分を加算して、現金書留にてお申し込み下さい。

MICROCOMPUTER SHOW & BUSINESS SHOW

マイコンショウ'90&第70回ビジネスショウ

例年どおり東京平和島にある流通センターでマイコンショウ'90，晴海国際見本市会場で第70回ビジネスショウが開催された。いずれも今後の世の中の動き，特にコンピュータ業界の趨勢を占う重要な催しといえるだろう。

今年から来年にかけての業界や新製品の動きを先取りする両ショウの内容をパソコン関係を中心にレポートしてみたい。

マイコンショウ

❶ずらりと並んだカラー液晶
❷ファミコン程度の絵なら小型液晶でも十分鮮明。反応も速い
❸鮮明なTFTカラー液晶ディスプレイ
❹これがマルチワード
❺DATメモリ
❻ICEを使ったゲーム開発
❼ついに実物が見られた68040
❽V70とAFPCのボード

ビジネスショウ

⑨最近のオシロはとにかく凄い（ソニーテクトロニクス）
⑩これがR800
⑪電波新聞社。開発中のギャラガ'88が見える
⑫TRONは元気だ
⑬TX1を使ったラップトップB-TRON
⑭なぜかJALもTRON。ウィンドウにはビデオが映る

マイコンショウ

年々地味な様相を呈するマイコンショウ。大手メーカーではパソコン，液晶ディスプレイやカスタムLSI，ワンチップマイコン，各種コントローラなど，その他サードパーティによる開発システムや周辺機器が主な出展物だ。

シャープでは業界最強のカラー液晶を始め，大小充実した液晶ディスプレイ，エレクトロルミネッセンスディスプレイなどを展示。もはやお馴染みとなった14インチ液晶カラーテレビも，いまだ他社の追随を許していない。しかし，他社の追い上げも激しく，今回の出展では日立の10インチカラー液晶ディスプレイユニットが注目株だ。

X68000関係はX68000 SUPER-HD，SX-WINDOWが中心。昨年登場していたWORD PRO-68KはMultiWord と名前を変え，内容を一新して出展。高速テキスト画面と印刷イメージに忠実なウィンドウ画面を高速に切り換えて使用する。

CPU関係では，なんといってもR800。

R800はアスキーとヤマハの共同開発によるZ80と上位互換性があるという16ビットCPU。同クロックでもZ80の3倍以上の速度を持ち，乗算命令を追加，アドレス空間は16Mバイトという，Z80ファンなら思わずスキップしたくなる仕様。ちなみに最高29MHzで動作する。うーむ，MSX3でも出ないかな～。

こういったショウでは初めてモトローラの68040が展示されていたり，NECの国産32ビットCPU，V70/80シリーズ用のAFPC（高性能浮動小数点コプロセッサ）がようやくお目見えしたりと，見る人が見れば期待のプロセッサ群が発表されていた。

そのほか，最近のFAは凄いと思わせられたのがソニーテクトロニクスのオシロスコープ。カラー画面を採用し，複雑な操作も画面のメニューに触るだけ。CRTに張られたタッチセンサと指の動きを捉える光センサのマトリクスが決め手だ。ベンツ1台分の値段というのが残念だが，家電製品にもこんな操作性があればと思わせる。

ビジネスショウ

SIS（戦略情報システム）の文字が目につくビジネスショウ。文具類は電子文具に代わり，オフィスの電子化はまだまだ進みそうだ。

❶これがAll in Note
❷ファクシミリもフルカラーの時代
❸電子手帳の住所録をハガキに印字する専用プリンタ
❹ショウでは必ずこういうのもある
❺ずらり並んだICカード
❻Macintosh Ⅱ でビデオフロッピーを読む
❼いち早くMachと国際化MOTIFを採用したLUNA88K
❽大きく見えるが本当は小さいラップトップLUNA
❾これがラップトップNEWSだ
❿X-Windowでテレビが映る
⓫噂のパームトップ。これも68000マシンだ
⓬8cmCD-ROMのデータディスクマン
⓭XEROXの両面レーザープリンタ
⓮AXにつながったDVI
⓯これでオフィスから煙草の煙を追放できるかな？

コンピュータ関係を見るとラップトップパソコンは当たり前。今年は実用レベルでのカラー化ほか，各社とも A4 以下のブックパソコンを出展して注目を集めていた。

そんななかでTFTカラー液晶を使ったシャープのラップトップAXの鮮明さ，そして8階調液晶と12MHzの80C286を擁し，さらに20Mバイトのハードディスクを内蔵して重さ2kg以下というAXパソコン，All in Note（MZ-8376A）の小ささが目をひく。

特にAll in Note。大きさはA4よりひとまわり小さいフリートップサイズで，Dyna Book比60％以下の容積で収まっている。補助バッテリユニットをつけると奥行きが少し大きくなるが，その場合は5時間の連続使用が可能となる。

もっとも小さく，軽く，大容量で性能的にもPC-9801NSに次ぎ，しかも大画面。

家電も含めてシャープは昔から製品を小さくすることでは定評があるが，とにかくこれで，やっと本当に使えるブックパソコンが出てくれたという感じだ。

さらに，ワークステーションもラップトップ時代に突入。デスクトップサイズになって驚いていたのが嘘のような勢いだ。小さくても内容はパソコンとは一線を画す。メインメモリは当然メガ単位，液晶やプラズマの高精細ディスプレイ，100Mバイト単位のハードディスクを内蔵する。ただ，用途はいまいちはっきりしない。

マルチメディア関係ではCD-ROM が元気。実績で先行する富士通&ソニーのCD-ROM XAとインテルの推す DVI などが出展。CDI はみつけられなかった。画像圧縮では一歩先を行くDVIはCD-ROM 1枚に70分以上の動画像を収録する。デスクトップビデオの夜明けは近いか？

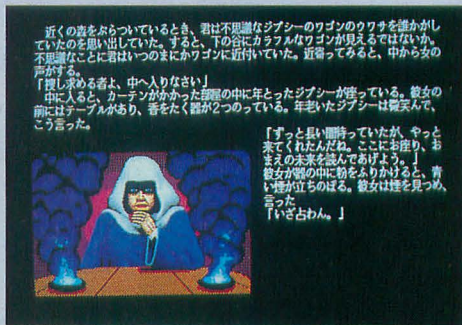
そのほか，ソニーはデータディスクマンを展示。これはCDシングルサイズの CD-ROM専用端末だ。標準装備のCD-ROM 1枚に数種の英和辞典と和英辞典，国語辞典，漢和辞典などをまとめて収録している。これだけでも十分実用になるが，もっともっと可能性は秘めている分野だけに，今後の展開が注目される。ディスクマンだから，もちろんCDシングルの音楽も聞けるぞ。

さて，今回のショウでは会場に女性の姿が目立ったのが印象的だった（無論，コンパニオンではなく）。嬌声をあげて大江千里のパンフレットを掻き集めるOL。ビジネスショウも変わった……。 （S.N.）

THE SOFTOUCH

SOFTWARE INFORMATION

プロミストランド
あのポピュラスのシナリオディスクです。フランス革命，江戸時代などの歴史に沿ったゲームが展開できます。

ウルティマV
いわゆる悪者と戦うことから離れ，人間の深層心理を舞台にしたRPG。心のなかの正義と悪との葛藤を描いている。7月21日発売予定。

SOFTWARE INFORMATION

さて，今月もいろいろなソフトが出揃いました。すでに発売されているものもあるので，雨の中いまから買いに走ってあとは家にずーっとこもるという大技に出るのもいいかも(ただし時間が許せばだけど)。

話題のソフトウェア

あー，うっとうしい！　何がうっとうしいかって？　決まってるでしょ，雨よ雨，梅雨の雨。まったく，この雨のおかげで靴はぐちゃぐちゃになるし，傘は持たなきゃならないし，持病の喘息発作は起こるしで，いいことなんかありゃしない！　あーあ，早くこの雨やまないかなー（なんて書いて梅雨になってなかったらどーしよー，何を隠そういまはまだ5月なのだし……)。

さて，この雨を蹴散らすかのように，今月もたくさんの新作情報が集まりました。

まずはこの**プロミストランド**からいきましょうか。いまや人気絶頂のリアルタイムシミュレーション，ポピュラス。つい先日，編集部でも2時間15分にわたる西川善司VS祝一平の対戦が行われたほど。まあ，この結果はまた後日お伝えするとして，今度はそのポピュラスのシナリオが登場するわけです。江戸屋敷ステージほか5つの舞台が設定されています。これでまた飽きることなく，新たな楽しみが味わえそうですね。発売はイマジニアより7月6日の予定。

お次はポニーキャニオンから発売の**ウルティマV**。お馴染みアメリカで大ヒットしているRPG，ウルティマの最新作です。ずっとX1でシリーズが出されていたので一瞬喜んでしまった方，ごめんなさい。今回はX68000での発売なんです。で，今回の諸悪の根源は，なんと心の影の化身シャドー・ロード。そうです，今回は人の心の“正義と悪”がテーマなんです。何が正しくて何が悪なのか，それを探し求めて再び聖者アバターはブリタニアを旅する，といったストーリー。操作性もグンとよくなったし，

ついにポピュラス王座へ！

順位	タイトル	前回
1.	ポピュラス	2
2.	ダンジョンマスター	1
3.	ワンダラーズ・フロム・イース	3
4.	グラナダ	—
5.	三国志II	12
6.	サーク	—
7.	ソーサリアン	7
8.	アルガーナ	5
9.	ファーストクイーン	8
10.	ジェノサイド	6

ポピュラスが着実に票を伸ばし，トップの座を奪いました。長く遊べるタイプのゲームは一旦トップを奪うとしつこいですから，しばらく居座るんじゃないかな。夏には，新シナリオを携えたダンジョンマスターを「プロミストランド」で迎えうつことになりそうです。

イースファンの皆さんにはすみませんが，今月も「ワンダラーズ〜」は3位でした。「ファルコムにソフトを出させるためにも，みんなで買おう！　目標5万本！」なんてハガキもあって，1枚1枚に相当力が入っているんですけどね。でも，イース I・IIの発売は○○からなんですよ。ナイショだよ。

チャートのまんなかを見ると，グラナダとサ

天下統一

REINFORCER

ジェミニウイング

実戦ビリヤード

ユニオン

なかなか考えさせてくれて楽しめそう。

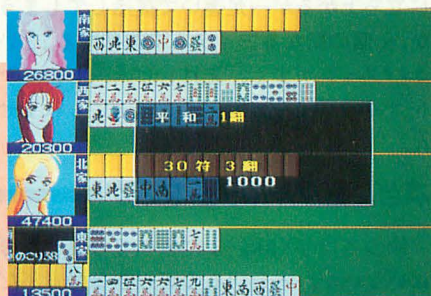
そしてシステムソフトからは**天下統一**がいよいよ発売となりました。本格派の戦国シミュレーションですが，難しく考える必要はまったくナシ。ただ，日本史に興味があれば，より楽しめるでしょう。さあ，みんなで天下統一を目指そう。

次々と新作を出してくれているザイン・ソフトでは，ただいま**REINFORCER**というシューティングゲームを開発中。詳しいことはまだお伝えできないけど，とりあえず画面写真が届いたのでお見せしちゃいましょう。このほかザイン・ソフトでは，グラフィックツール**Gツール**も開発中です。

さて，システムサコムの**ジェミニウイング**ですが，着々と進行しているもようです。とりあえずデモソフトが届きましたのでお見せしますね。発売は7月中旬の予定です。

パック・イン・ビデオでは，**実戦ビリヤード**を移植開発中とのこと。これはPC-9801で人気だった同ゲームをさらにバージョンアップしたもの。ナインボールのほか，ローテーション，四つ玉，スヌーカーなども楽しめる本格派です。

新規参入のポニーテールソフト/STUDIOオフサイドでは，麻雀牌などを使ったパズルゲーム**ユニオン**を開発中。これはステージ上におかれている牌を，同じものどうしで合体させていくというもの。牌は動物，くだもの，昆虫などがあって好みで選べます。発売は6月末から7月上旬の予定。

さてさて，タケルソフトからは**ぴんぴん麻雀ピーチエンゼル**が発売中。お相手を務めるのはもちろん美女。お楽しみが見られるかどうかは腕次第ってとこかな。

スタジオパンサーからは**天九牌Special 桃源の宴Part 2 女子高生編**がタケルより発売中です。これは，天九牌のなかの桃源境編を抜き出し，ボーナスステージや女の子を増やして強化したものです。

タケルといえば，LOGINのソフトウェアコンテストの優秀作が3本発売されています。RPG**風神魔伝II**とシューティングゲーム**PLANET**，パズルゲーム**フェブリー**がそれ。2,000円ならお買い得ってカンジです。

そうそう，光栄からX1turbo用シミュレーション**大航海時代**が出ました。世界を回り最高爵位を目指すゲームです。

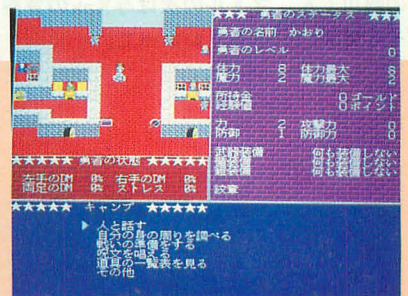
そのほか，レイトレーシングソフト**C-TRACE**がバージョンアップされて発売されます。計算時間のスピードアップ，ツリー構造の採用などさらに使いやすくなって登場です。6月下旬発売予定です。では，来月。

ぴんぴん麻雀ピーチエンゼル

風神魔伝II

ークの新鋭が揃っています。サンプルから製品になって見違えたグラナダはまさに追いこみの勝利。サークはビジュアルへのこだわりが，X68000ユーザーの胸に響いたようです。これ以上ランクアップできるかどうかは微妙なところですが。

この2つに挟まれて三国志IIも返り咲き。なんてったってX68000にもない新作。X1ユーザーのみんな，一緒にいばろう。やーいやーい。それにしても，X1用が3作もランクインしてるあたり，X1ユーザーのパワーも馬鹿になりません。

さて，来月グラナダは三強の一角を崩せるか？X1ユーザーの意地，三国志IIの動向は？（浦）

PLANET

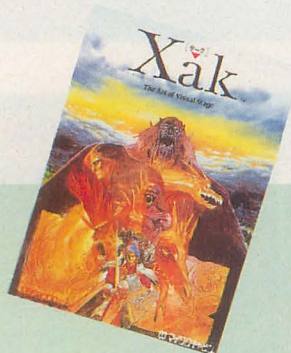
天九牌Special桃源の宴 Part2

THE SOFTOUCH

●Xak

与Xakは木を切るか

Nishikawa Zenji
西川 善司

他機種ではもうおなじみの「Xak」がついにわれらのX68000にも登場。これを紹介してくれるのはイースIIIの冒険記を終えたばかりの西川くん。マドルの次は与Xakの冒険だそうです。……もういいって。

X68000用 5"2HD版4枚組 8,800円(税別)
マイクロキャビン ☎0593(51)6482

私が水爆頭の西川善司です。突然ですが私は3DダンジョンタイプのRPGが苦手です。XTALソフトの「ファンタジアン」は最後までプレイしたけれど，それ以後はまったくこのタイプのゲームに魅力を感じなくなってしまったのです（ファンタジアンは面白かったなぁ)。私はそもそもあのマッピングというのが大嫌いでありまして，まして「すり抜けの壁」やら「ワープ」「回転床」などが出てきてしまうと，もういっぺんにやる気が失せてしまうのです。

あと，このタイプのゲームに多い緊迫感のない戦闘シーンが私は嫌いです。あの「1．戦う　2．逃げる　3．魔法」というメニュー式のやつですよ。これってゲームをしているという気がしないんですよね。

そんなわけで，私は大人気のビクター音産の「ダンジョンマスター」も食わず嫌いでいまだにプレイしていません。それに対して，この「Xak」のようなアクティブ・ロールプレイング（ARPG）は取っ付きやすくて面倒なマッピングもないし，きわめて私のような人種向きのゲームといえます。

さて，この「Xak」は去年の夏に発売されたPC-8801版がオリジナルです。思えば去年は各ソフト会社がARPGをこぞって出した年でありまして，ファルコムは「イースIII」，システムサコムは「プロヴィデンス」「ヴァルナ」，エニックスは「オールド・ビレッジ・ストーリー」，テクノソフトは「新九玉伝」，……と挙げればきりがないくらいです。ただ，ストーリーが「**高貴な血筋を引く主人公が，魔王の復活を阻止するために立ち上がるが一足違いで復活してしまう。が，x個の聖なるアイテムで封印する**」というパターンが多く，先月レビューしたファルコムの「イースIII」もこの例にもれません。

本当に私は神の子孫なのか

俺の名は与Xak(ヨサーク)。まだ気立ての良い女房はいないが，木を切るのが好きな好青年だ。北島三郎の歌とはこれっぽっちも関係ないのは先にいっておこう。

エリス「おじいちゃんがメガネをなくして困っているの。探すのを手伝ってくれない？」

与Xak「あ，いいけど」

このエリスっていうのは町長の孫娘で俺のスケっ，じゃなかった幼なじみだ。

エリス「じゃあ，教会を見てきてくれないかしら」

うおおぉおっ。ガンガンガンガン（頭をキーボードに打ちつけている音）いきなり初めっから使いっぱだぜ。こりゃ，先が思いやられるなぁ。で，眼鏡を持っていくと……。

町長「おお，待っておったぞ。お前の家を訪ねてはるばる南のお城からお客が来とるんじゃ」

メッセンジャーピクシー「あなたが戦士ドルクさんですか」

与Xak「ドルクは俺の親父だよ。親父は半年前に旅に出たきり行方がわからないんだ」

メッセンジャーピクシー「そうですか。戦いの神デュエルのいない今となっては子孫であるあなただけが頼りなのです（きた，きた，きたぞ～)」

与Xak「こ，この俺が神の子孫！　うおおぉぉ，ガンガンガンガンガン」

メッセンジャーピクシー「どうかしました？」

与Xak「神の子孫が，眼鏡探しをさせられたなんてあんまりだと思って……」

RPGの主人公は風呂に入るのか

武器を買い揃えたら町の外へ出よう。外にはスライムやスケルトンなどがいる。戦闘の方法は単純明快。「イース」や「ハイドライド」と同様，剣を構えて敵に向かって突っ込んでいけばいいのだ。ただし，真正面から突っ込んでいったのではダメージを受けやすいようだ。敵の背中，または側面を狙うようにしたほうがいいだろう。

また，ダメージを食らったら敵の来ない安全な場所で休もう。全角キーをONにしゲームスピードを上げてから休むと回復のスピードも速くなって気持ちいいぞ。

細い山道を通って東のほうへ進むと，「西遊記」の八戒のようなやつ（オーク）と赤い手長猿のような敵（トロール）の出現する場所に出る。レベル2になったらここで修業するといいだろう。

さて，その北には穴の中でゴーゴー（死語）を踊るワーウルフの森がある。レベル3になったらここがよい経験値稼ぎの場所となる。おや，あそこで寝そべっているの

おや，こんなところに女の子が倒れているぞ

は？ ……女の子だ。大変だ，ケガをして倒れてるぞ。
与Xak「ど，どうしました？」
フレイ「ケガをしてしまったの。家が近くにあるから連れていってください」
与Xak「よし，背中に乗りなさい」
フレイ「……くさい」
与Xak「ははは，そりゃそうだ。冒険に出てから一度も風呂に入ってないからね。『イース』シリーズの主人公アドルは3年入っていないそうですぜ。きっと奴の体はタムシと水虫だらけだろうな。西川善司でさえ，1週間に2回は入るのにねぇ」

フレイの家は彼女が倒れていた場所から東へ行ったところにあった。
フレイの父「おお，フレイ。無事だったか。あれほど水汲みはいいといったのに。これはひどい臭いと傷だ。与Xakさん，娘を町の病院まで連れていってくれませんか？」
与Xak「うおおぉぉ，ガンガンガン」（XF3キーがとれちゃった）
フレイの父「どうなされました？」
与Xak「あ，ただの発作です」

病院の看護婦はオレンジロードか

病院はフェアレスの町の南東の隅にある。
看護婦「あら，与Xak君。……どうしたの，その女の子は？」
与Xak「森でモンスターに襲われてケガをして倒れていたのを俺が助けたのさ。診てやってくれないかな。それよりあんた，『**きまぐれオレンジロード**』**の鮎川まどか**にそっくりだね」
看護婦「うふふ。MSX版の『Xak』の看護婦なんか『**めぞん一刻**』**の管理人さん**の音無響子そっくりだったんだからぁ」
与Xak「はぁ？ じゃあ，FM-TOWNS版は**ちびまるこちゃん**か**サザエさんのワカメちゃん**だなぁ，きっと」
看護婦＆与Xak「はははは」

……マイクロキャビンさん，こういう同人誌レベルのギャグは真面目なゲームに入れると逆効果だと思います。

そんな顔しないでよ，看護婦さん

森の奥にどっしりとかまえている木の精

木の精はレベルいくつで倒せるか

フレイを病院へ送り届けたことを親父さんに告げると，今度は森の木の精を倒してほしいという。木の精はワーウルフの森のいちばん奥にいる。が，レベルが7以上ないとまったく歯が立たない。そこで以下に各レベルで倒せるモンスターのうちでいちばん経験値をくれるやつを記しておこう。レベル11以降は自分で調べてちょうだい。

レベル4 バジリスク（トカゲみたいな奴）
レベル5 トレント（木の化け物）
レベル6 サイクロプス（一つ目の巨人）
レベル7 毒沼に住む赤い奴，サンドマン，Dファイター（砦にいる騎士）
レベル8 木の精
レベル9 ケルベロス（砦にいる犬）
レベル10水竜（砦のボス）

これを見てもわかるように，木の精を倒したあと行くべきところは「砦」だ。先は長いぞ，がんばれ。

Xak2は出るか出ないか

「Xak」はなかなかよく出来ていますが先月レビューした「イースIII」と比べてしまうと，ちょっとムムムの点があります。まず，スクロールがX68000なのに**16ドット単位**だということ。オリジナルのPC-8801版が，キャラと背景の重ね合わせをチップ単位で行っていたからといって，そのまんま移植することはなかったと思います。

まさか，この犬は火を吐かないだろうな

そして，やっぱりストーリーに新鮮味がないですね。結局最初にいったARPGのストーリーパターンそのまんま。大筋はそうなってしかたないとしても「イース」シリーズのようなドラマチックさがもっと欲しいところです。1つひとつのイベントに登場するキャラクターが使い捨て的に出てくるだけで，ストーリー全体を通しての伏線が張られていないのです。キャラクターが単なる「主人公にアイテムを渡す人」なんですね。だいたい**主な登場人物**と紹介されているヒロインのエリスとフレイがゲーム前半に出てきただけで，そのあと一度も出てこないのはどういうわけでしょう。

最後にゲームバランスについて。この「Xak」では，経験値稼ぎをさせられる局面によくぶつかります。新しいフィールドに移ると，そこに出現する敵にまったくダメージを与えられず死んでしまうことがよくあるのです。ボスキャラも同様です。そうなると前のフィールドに戻って弱い敵をレベルアップのためにえんえんと殺し続けなくてはいけないのです。この辺も「イース」シリーズを見習ってほしいところです。

おまけ：ゲームの最初のピクシーの頼みを断り続けたり，フレイをおぶったままエリスの家に行くとエリスの表情が変わる。

総　評

ソーヒョー！（釣りキチ三平のまね）

いいたいことは本文に書いたので，ここではBGMとグラフィックについて書きたいと思います。

「Xak」はBGMがとてもいいです。リズムはAD PCMでやっているし，鶏の鳴き声なんかのSEが入っちゃったりベースのソロなんかも入ったりして，ゲームのBGMとしてはとても斬新な感じがしました。ゲームミュージックファンは要チェックです。

一方グラフィックですが，店やイベントで表示される人物などの絵が最近ありがちなオタッキーなものとは一味違ったタッチで，色の使い方も綺麗で良いと思います（EXCEPT 病院の絵）。しかし，通常のゲーム画面の背景はいかにもチップで構成しましたという感じで「イース」なんかと比べると多少見劣りします。キャラクターもなんとなく寸詰まりですしね。この辺はもう一工夫するべきだったと思います。

ソーヒョー！（釣りキチ三平のまね）

グラフィック	7
BGM	9
ゲームバランス	5
ストーリー	6
操作性	7
ディスク周り	6
技術	6
総合点	7

●あーくしゅ

ウルフファン必須のパロディゲーム

Koga Kazunori
高河 和海苔

パソコンゲーム界に数々の名作を残してきたウルフ・チームが放つ，ファン必見のパロディアドベンチャーゲーム。いままでのゲームの主人公たちがかわいくデフォルメされて，画面のなかで大活躍するぞ。

X68000用　5"2HD版3枚組　6,800円(税別)
ウルフ・チーム　☎03(5273)4795

ドンドンドンドン，と無言のノック。この春もまた新聞勧誘員のせいで，とても不愉快な思いをしました。

ドアを開けると，靴の先をドアの隙間に差し込み，不当景品をちらつかせながら，「いまどこの新聞をとってるの？」と，いつものセリフ。知ってるくせに！　ま，その程度ならまだしも，「俺を誰だと思ってるんだ。玄関のドアがなくなってもいいのか」だとか「そこのフライパンで殴ってやろうか」だとか言われたこともありました。その夜からは数日間おびえながら生活していた小心者の私です。

思いあまって某大新聞社に直訴の手紙を出したこともあったのですが，「そんな末端の事情は知ったこっちゃない」ってお返事でした。ああそうですかあ。

とにかくかわいい～

そんな憂鬱な気分も和らぐ「あーくしゅ」を紹介しましょう。ハッキリ言って，遊んでる間ずーっと面白かったゲームというのはひさしぶりです。ニュージーランドストーリーに次ぐ，私のおすすめ品にしましょう。しかし，独創的なシステムだとか，きわだって優れたシナリオだとか，そういう面からの評価はできません。普通のコマンド選択式アドベンチャーであり，アイテムを集めて問題を解決させるんだタイプのシナリオだからです。じゃ，何がよかったのかというと，グラフィックと会話，ほのぼのした雰囲気が気に入ったのです。とにかく，かわいい。3等身にデフォルメされたキャラクターたちがとってもかわいい。じぇだやスーニーの大ボケな言動がかわいい。サーラのムチがいとおしいってなもんです。

まあ，「買い－捨て」の判断は人によってまちまちで「出来がいい－悪い」だったり「メジャー－マイナー」だったりするかもしれませんので，なにがなんでも無条件におすすめとまでは言いませんが，「かわいい－かわいくない」を評価の基準に据えている方々には，ぜひ楽しんでもらいたい作品だな，と思います。

パロディなんです

「アークス」を舌っ足らずに発音したタイトルからも想像できるとおり，あーくしゅはアークス，およびアークスIIのパロディになっています。

そのほかにも，ちまちまとした戦闘の「ファイナルゾーン」や，渋さが売りものの「ヤシャ」，優子と麗子の「夢幻戦士ヴァリス」，いまひとつ知名度の低い「ガウディ」，壮大なストーリーの「ミッドガルツ」，凝ったオープニングが評判の「斬」，最新作「グラナダ」，といったゲームからも，キャラクターたちが参加しているという，いわばウルフ・チームから生まれたキャラの大パーティなんです。

そんなわけで，元ネタを知っているほうがより楽しめるあーくしゅですけど，知らなくても大丈夫です。少なくとも知ってなければ困る，解けないなんてことはありませんので，安心してプレイしてください。

聖剣と仲間を探すのだ

さて，主人公はアークスから引っ張ってきた，じぇだとピクトです。この2人が一緒に冒険をするわけですが，なぜ2人組なのかっていうと，じぇだだけだと話が進まないし，ピクトだけだと面白くないからなんでしょうね，きっと。ピクトが常識をわきまえた好青年なのに対して，じぇだは脳天気の大ボケ野郎で，なにかといえば「おなかすいたね～」「おべんと～」「ほう」なんて言ってる無邪気な22歳なんですから。まったく，かわいいぞ，こいつぅ。

それではゲームスタート。まずはどこかの田舎道からです。2人はぼんやりと旅を続けていたようですが，少し歩くと金竜リグ・ヴェーダに出会い，時間の秩序を守るために協力してくれ，と頼まれます。

いきさつを簡単にお話ししましょう。

むかーしむかしから，3本の聖剣というものが存在していて，これらがすべての時と空間を支配していました。ところが，いろいろな時代を行き来していたものですから，いくつかの次元に裂け目ができてしまい，世界が混じり始めてしまったのです。このままでは，時間の流れが乱れてしまいます。この裂け目を埋めるためには，3本の聖剣を合わせたエネルギーが必要です。裂け目を通っていろんな時代に行き，聖剣を集めてください。3本の聖剣はそれぞれ違った音楽に反応してその姿を現すので，まずは3枚のCDを探してください。と同

冒険のはじまりはじまり～

時に，裂け目から吸い込まれてしまった仲間たちもこの時代に連れて帰りましょう。では，よろしく。ってな具合です。

この役目を請け負った2人は，次元の裂け目に突入，そこにはなぜか，U子と0子(優子と麗子ね）が店員をしているDOSバーガーがあるのでした。

0子　「いらっしゃいませ，こんにちは」
じぇだ「はんばぁぐ」
0子　「はい？」
じぇだ「ぽてとぉ」
0子　「あ，すみませんもう一度……」
じぇだ「こぉら」
0子　「あ，あのぉ……」
じぇだ「ぴ〜くと〜ぉ，か〜えな〜いよ〜お」

まったく，じぇだはまともに会話もできないんだから。ほかにルアン・カーンっていう，どこにでも出てくる変なのもいます。

ピクト「すっごい雲の波だねー」
じぇだ「すご〜い」
ルアン「ははははは！　愛と勇気の男ルアン・カーン参上！」
じぇだ「すご〜い，くものうえにたってる〜」
ルアン「見よ！　この勇姿！　すべての女性を魅了する甘いマスクと強靭な肉体を兼ね備えた光の騎士ルアン・カーン！　ふははは，世界は私のためにあーり！　ふはははははははは！　わぁー！」
ピクト「落ちた……」

そんなこんなで，くだらないだじゃれが終始ちりばめられつつ冒険はなごやかに行われます。で，もちろん超未来，近未来，現代，過去を駆け回って，3本の聖剣と6人の仲間（エリン，グラン，スーニー，バザン，サーラ，チノップ）を探すという本来の目的も大切ですが，このゲームではそこまでの過程，1つひとつの会話もじっくり楽しんでほしいと思います。

ちなみに，エンディングはアニメ番組のそれっぽくて，いかにもふさわしい感じのものでした。

次元の裂け目を通ると，そこは……

親切な設計

最近はゲームオーバーのないアドベンチャーゲームやロールプレイングゲームが増えています。このあーくしゅも一応ゲームオーバーがあるのですが，実はないのとたいして変わりません。

それはガウディにもあった，ヒストリーリピートシステムの搭載によります。ゲーム中であれ，ゲームオーバーになったときであれ，画面上にあるHRっていうアイコンをクリックすると，過去20画面分が縮小されてずらずらっと表示されるのです。この中からどれかをクリックするだけで，そこからやり直すことができます。これは便利。もちろん普通のセーブ機能もあります。

それから細かいことですが，マウスカーソルの形が矢印や指ではなくて，セーラー服の女の子なのが泣かせます。こうした遊び心は大歓迎ですね。

おまけに，ミュージックモードもついていて，音楽を聴くだけでなく効果音を聞いたり，グラフィックを見たり，スタッフからのメッセージを読んだりもできます。BGMはいままでのゲームをアレンジしたものですが，音色には相変わらずのパラメータを使っているようです。

フルマウスオペレーションといい，十分な処理速度といい，完成度の高いシステムに仕上がっているといえるでしょう。

しかし，多くのアドベンチャーと同様に，Aから手にいれたBをCに渡すとDをくれるので，それをEに渡すとFに案内してくれて，その場所でGで買ったHを使うとIとJとKが見つかるので云々，と不条理なつながりで，ストーリーが進行していく事実は否めません。パズルゲームのようで，面白いといえば面白いんですけどね。

最後にヒント

あーくしゅはすでに市販されているので，

江戸時代まであるとは

ミュージックモードもまた楽し

もうとっくに解いちゃったよ，っていう人もいることでしょう。が，このテのゲームに慣れてなくて，ハマってしまっている人もいるかもしれません。そんなわけで，ちょっとヒントでも書いておきましょうか。

だけど，これから遊ぼうと思っている人は以下の文をゼ〜〜〜ッタイに見ないように。ではでは，がんばろう！　ほう！

- お金を貯めるには，超未来に出てくる敵Mの顔を誉めるのが手っとり早い。
- 眠っているEには，彼女の好きなものをさらに飲ませてあげよう。
- Sはムチの置いてある店にいるので，しつこく探してみよう。
- 光の浪人Rなんかに大切なモノを渡したくはないが，見返りが期待できる。
- 過去はしばしばうろついてみること。
- 封印のされた洞窟を開けるためには，Sを同伴していなければならない。

<総評>

ゲームの購入の選択は，お金のあるなしと，時間のあるなしによって，結構変わってくるものかもしれません。

時間があるのにお金がない人は，光栄の歴史3部作のようなSLGなどを買って，じっくりと楽しみたいと思うのではないでしょうか。また，時間がなくてお金がある人は，テトリスやアフターバーナーなどを買いそろえて，気分転換にちょっと遊んでる，なんてぜいたくをしていることでしょう。

あーくしゅは，日曜のお昼に始めたら，夕方には終わってしまいそうなお手軽アドベンチャーですから，特にこんな方にはぴったりだと思います。

なお，時間があるうえにお金もあるお方，および時間もないくせに金もない！　って方に対するコメントは控えさせていただきます。礼。

評価表（最高は♥10個）

項目	評価
グラフィック	♥♥♥♥♥♥♥♥♥♥
シナリオ	♥♥♥♥♥♥♥
サウンド	♥♥♥♥♥♥♥
完成度	♥♥♥♥♥♥♥♥♥
少女趣味	♥♥♥♥♥♥♥♥♥♥
ほのぼの	♥♥♥♥♥♥♥♥♥

THE SOFTOUCH

●ダウンタウン熱血物語

ストレス解消用ケンカアクション

Yamada Junji
山田 純二

ファミコンからの移植でおなじみ「熱血硬派くにおくん」シリーズ第3弾。今回はくにおくんのライバル，りきが主役って感じです。さてさて，いったい何が起こることやら……。

ファミコンで人気のダウンタウン熱血物語が，シャープから発売になりました。お馴染みSPSによる移植ですが，移植の完成度はもちろん完ペキ，元のゲームの味をそこなうことなく，元気たっぷりのアクションが楽しめます。

ゲーム内容は，群がる不良どもをバッタバッタ薙ぎ倒し，相手から巻き上げたお金を使い，商店街で暴飲暴食をして（なんてヤツだぁ！），どんどん強くなっていくという，ストレス解消にはもってこいのアクションゲームです。とはいえ失敗して「ふくろだたきの刑」にあうと，逆にストレスが溜まりますけど。

このゲームでは，熱血高校ドッジボール部でお馴染みのくにおくんともうひとり，くにおの永遠のライバルである花園高校番長りきの2人によって，ゲームが進行していきます。ひとりでも2人でもプレイは可能で，2人プレイではくにおとりきが互いに反発しあったり，協力したりという2つのモードが用意されています。仲のよい友達とプレイする場合には協力モードで，嫌な野郎とは反発モードで（でもわざわざヤなヤツとはやらないか）遊んでみるといいかも。ちなみに，この2つのモードの違いは，プレイヤー同士の当たり判定があるかないかだけです。

ひとりの場合は，りきかくにお，どちらかを選んでプレイするんですが，ストーリーを読んでみると，今回の主役はどうやらりきのよう。くにおくんは単なるオマケのような感じで，くにおくんが好きな私は，ちょっと寂しい……などとぼやいていてもしょうがない，そろそろりきのお手伝いでもしに行きましょうか。

運命の再会

花園高校，校門前。怒りながら走ってくる人影と，校門前に立っている人影が見える。

くにお「りき，どこへ行くつもりなんだ」

りき「く，くにお！　おまえ，なにしに……」

くにお「冷峰学園へ行くんだろう。ちょっとつきあわせてもらうぜ」

りき「きさま，どうしてそれを知っているんだ！　まさか，今回の事件に，お前も絡んでいるんじゃないだろうな」

くにお「残念ながら，関係ないな。どうした，行くのか，行かないのか」

りき「勝手にしろ！」

今回はなんと一大事！　最近ハバをきかせ始めた冷峰学園の生徒に，りきの彼女がさらわれてしまったのだだだ。そして，りきの手元に届いた1通の挑戦状……。「彼女を返してほしければ，冷峰へこい」。

こうして，2人は冷峰学園へ向かったのでした。

冷峰学園ってどこ

りき「ところで，おまえ冷峰学園がどこにあるか知ってるか」

くにお「……まさか，おまえ知らないって言うんじゃないだろうな」

りき「そりゃあ，挑戦状に書いてなかったし，あんな弱小学園，いままで気にかけたことなんかないからな」

くにお「おめでたい野郎だぜ……」

なんとも間抜けなお話，挑戦状を受け取ったはいいが，2人ともかんじんの冷峰学園の場所を知らなかったのです。このあと何度か罵りあった2人，結局は地道に冷峰を探すことにあいなりました。

さて，冗談はともかく，マップの繋がりはかなりややこしく，空間が歪んでいるとしか思えないほど（！）の場所があるので，自分の方向感覚に自信がない人は，マッピングしていきましょう。ちなみに私は，ちょうど暇だった編集のA氏をひきずりこんでマッピングをやらしてしまいました（おいおい）。

ゲームを始めて，まずたどりつくのが花園商店街。そこを抜けていくと花園第三公園へたどりつきます。ここらへんから，周辺の高校生の不良どもが現れるので，ようやくゲームらしくなってきます。

くにお「おい，坂宿高校のやつらがやってきたぜ」

りき「ふん，あんなやつらなんかどうって

X68000用　5″2HD版2枚組　8,800円（税別）
シャープ　☎03(260)1161

うっ，目が飛び出るほどの木刀の痛さ！

ことないさ」
くにお「強がってんじゃねえよ」
りき「なんだとこの野郎！」
と言い争っている２人を襲う鉄パイプ２本。スコーン！
くにお「いってー，あんの野郎よくもやってくれたな，ただじゃおかねえ」
りき「俺様の実力を思いしらしてやる。かくごしやがれ！」
ん～，まさに熱血，単細胞。まったく，頭に血が上ると見境いなくなるんだから，この２人は。てなわけで，アクションゲームの本領発揮です。このゲームでは，手に取れるものは全部武器となります。たとえば，敵さんが残していった木刀，ナックル，チェーン，そして鉄パイプ，さらには道端に転がっているポリバケツ，タイヤ，石ころ，古本，きわめつけは気絶して倒れている人間さえも，持ち上げてぶん投げることができます。特に，チェーンを振り回しているときなんて，気分はもう「おーほっほっほ，女王様とお呼び！」の世界，う～んチェーンの唸る音がとても気持ちいいっ（あっぶねエ）。

ところかわって桜町

なにやら薄汚い工事現場。敵を倒しそのまま進もうとしたら，どこからともなくディスクアクセスの音がして沢口君の登場。
沢口「ちょっと待ちな！　俺はおまえらのようなやつらを通すなと，西村さんに言われている。ここを通りたかったら俺を倒すんだな」
くにお「俺たちは，てめえなんざにかまってるヒマはねえんだヨ，怪我しないうちに下っ端はとっととすっこんじまいな」
りき「そうそう，冷峰学園の場所を教えてくれたら見逃してやってもいいぜ」
沢口「なっ，しゃらくせえことほざいてんじゃねえ！　ぶっ殺してやる！」
などという会話があったかどうかは知らないけど，戦闘開始です。ここは足場が狭いので，工事現場に落っこちないように注意。先手さえ取れれば，たいした相手ではありません。マップのあちこちには，このように中ボスが存在します。なかには有益な情報をくれたりするやつもいるので，余裕があれば逃げずに戦いましょう。

お店ではいろんなものを買えるのです

くにおと長谷部，感動の再会!?

しゃべるんですう

ところ変わって，ここは新宝川花園大橋。くにおは中学時代の同級生，長谷部と再会することになる。
長谷部「ひさしぶりね，くにお♡♡♡♡。冷峰の長谷部さんが，とってもいいこと教えてあげる♡♡♡」
わあ，びっくりした。いきなりサンプリングでしゃべりだした長谷部さん。２行しゃべっては，ディスクアクセスしながら，えっちらおっちら話してくれた情報によると，りきの彼女をさらったのは，冷峰のダブルドラゴン竜一，竜二であることと，彼女は校舎のいちばん上のいちばん右の教室に，閉じ込められているらしい。そして，セリフの最後には，
長谷部「あたしは，冷峰の生徒だけど，くにおを応援するわ♡♡♡♡♡♡♡。絶対負けないでね♡♡♡♡♡。約束よ♡」
セリフの間のハートマークの連打。くにおくんも，あまりの迫力に圧倒されている様子。こころなしか顔が青ざめているようで……。
りき「たいへんだな，おまえも」
くにお「お互いさまだろ」

銭湯に入るとプリティなオシリが見られる

こんな軽ワザ師のようなマネもできちゃう

どうやら，りきのほうも彼女とは苦労している様子。妙にしみじみとしてしまった２人，なにをやってるんだ！　まだまだゲームを始めたばかりじゃないか，こんなところでめげてどうする，冷峰へのみちのりは遠いぞ，がんばれくにお，そしてりき！

＊

とまあ，こんな具合にゲームは進んでいきます（登場人物の台詞はちょっと違うけど）。このあと，りきとくにおは，緑町，白鷹町，星の丘町，そして夢見町をかけめぐり冷峰へ到達しますが，冷峰学園の門は閉ざされていたのです。冷峰の四天王を倒さない限り，中には入れないようです。四天王のうち望月と平は倒した，残りの２人はどこにいるのだろう……。まだまだ先は長いようです。きしむジョイスティック，うなる必殺技，盛り上がるBGM，遊び心も一杯のこのゲーム，大作とはいいがたいが，とても熱くなれるゲームです。

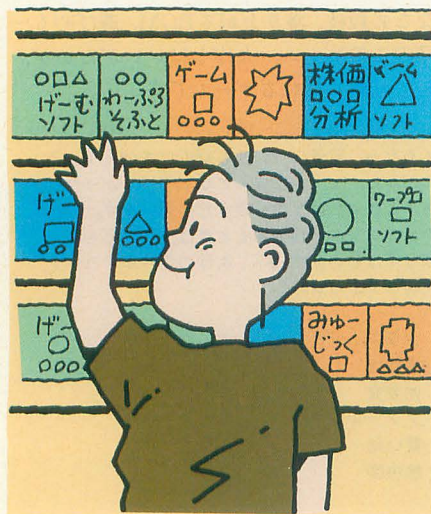
感想は……

このゲームでいちばん楽しいのは，食事のときのくにおくんのしぐさが（かわいい！）この一言につきます。もちろん，かっこいいアクションシーンも捨て難いが，くにおくんのいやいやにはかなわない，と僕は思います。ファミコンからの移植とあってゲーム性，バランス，ともにいいのですが，背景のグラフィックまでそのまんま使い回しが多いので，X68000でやっていると結構違和感を感じます。効果音については，サンプリングを使っているだけあって，文句ありません。ただ，一言文句を言わせてもらえるなら，背景についてもX68000を使っているんだから，もう少しなんとかしてほしかったなあ。この点だけが唯一残念だと思います。次にファミコンから移植するときは，ぜひともがんばってほしい。SPSさん，次作のワールドコート期待してまっせ。

グラフィック	7
操作性	9
効果音	10
アクション	9
買い物	8
熱中度	8

AFTER REVIEW

AFTER REVIEW

今月はグラナダ，FAR SIDE MOON，サーク，三国志IIの4本を取り上げてみました。なかなか皆さんやり込んでいるらしく，実際にプレイした感想がひしひしと伝わってきます。こういったハガキがどんどん増えてくれると，うれしいですね。

グラナダ

▶とにかく全面ぬかりなく，面白さがつめ込まれてて，感動することうけあい。演出がGood！

山形県・石澤　敏博(18)

▶サウンドにひどくひかれて買いました。でも，ゲーム内容はサイコー！　アーケードをこえています！　静岡県・小澤　健一(16)

▶BGMと多彩なオプション，デカイボスなどなど，カッコよかったから。

岡山県・加藤　雅浩(21)

▶ボスを見ると，オッと思わせてくれるから。

長野県・小笠原　信夫(15)

▶爆発音が仕事の苦しみを忘れさせてくれる。

東京都・山田　慎也(20)

▶難易度が変更できて誰にでも遊べる。

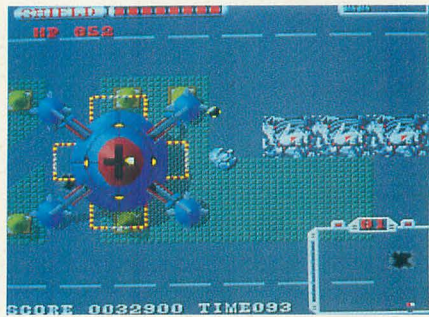
神奈川県・光石　和弘(17)

▶地味だが長く遊べるよいゲーム。

茨城県・石川　満章(22)

▶撃って撃って，撃ちまくりゲームだから。

神奈川県・丸藤　俊之(21)

▶バランスがよく，サウンドもグラフィックもよい。　東京都・井上　正男(17)

▶2面の空のステージは，とくに面白い。もちろん落ちるとちーさくなってくれるし，4門の誘導ミサイルもとーってもいいと思う。が，ボスキャラには神風にならなければいけないのが，すごく悲しい。

香川県・大西　力(18)

▶グラフィックにメッセージを感じる？　変なビジュアルシーンがないのもよい。ただ，自機がチャチいのが難。

三重県・長井　まさひと(24)

▶タンクゲームマニアがよだれをたらしてやりまくる気がするから。

茨城県・秋山　真一(16)

今回は，このグラナダに関するハガキがいちばん多かったですね。タンクタイプのゲームだけに，なんとなく古めかしいイメージを抱いてしまう人もいるかもしれないな，なんてちょっと心配してしまったりしたのだけれど，そんな心配はまったく無用だったようで，皆さん楽しんでいるようですね。なにを隠そう編集部でもこのグラナダは人気のマトで，来るたびに遊んでいくライターもいます。

最初は，操作性にちょっととまどうこともあるかもしれませんが，慣れてくると快適そのもの。ガンガン撃ちまくって気分も壮快。効果音も気分を盛り上げてくれますし，自分の腕に合わせてモードを設定できるので，誰でも楽しめます。しかし，まだMANIAモードでクリアした，という話を聞いたことがありません。お心あたりの方はぜひご一報を編集部まで。

X68000用　5″2HD版4枚組　8,800円
ウルフ・チーム　☎03(5273)4795

発売中のソフト

★天下統一

黒田幸弘氏が手がけた本格派戦国シミュレーションゲーム。プレイヤーは，戦国時代の英雄となって一城の主からスタートし，あるいは武将をしたがえ内政をほどこし，あるいは諸国の群雄を相手に野戦・攻城戦を繰り広げる。国を手中に治めたら，目指すは京，そして天下統一だ。今回移植を担当したのは，アルシスソフトだ。

X68000用　5″2HD版2枚組　9,800円
システムソフト　☎092(752)3902

★天九牌Special 桃源の宴Part2女子高生編

すでに発売されている「天九牌」から桃源郷編のみを抜きだしたのが「天九牌Special 桃源の宴」。これはその桃源の宴シリーズの2作目。こんどは，女子高生9人がプレイヤーのお相手を務めます。女の子を破産に追いこんで，お金を貸してあげればさらに増えたグラフィックが楽しめますぞ。

X68000用　5″2HD版2枚組　3,800円
ブラザー工業(TAKERU)　☎052(824)2493

★風神魔伝II

LOGINソフコン入選作のRPG。中世のたたずまいを持つフローランシア王国は，豊富な水と広大な大地に恵まれた平和な国だったが，突然現れた風神が国々を荒しまわりはじめた。この風神を倒すため，主人公のキミは立ち上がった。操作性やグラフィックもなかなかよし。

X68000用　5″2HD版　2,000円
ブラザー工業(TAKERU)　☎052(824)2493

★PLANET

LOGINソフコン入選作。全7面のシューティングゲームだ。海，山，宇宙空間と豊富に設定されたステージを5種類のパワーアップを使って突き進む。各面の最後には巨大なボスキャラとの対決が待っている。

X68000用　5″2HD版　2,000円
ブラザー工業(TAKERU)　☎052(824)2493

★フェブリー

これもLOGINソフコン入選作。コースの上に置かれた鉄球を転がしてゴールへ導いてやればよいのだが，コースから落ちないようにするためには，磁石を置いてうまく誘導してやる必要がある。全50面の新感覚パズルゲームだ。

X68000用　5″2HD版　2,000円
ブラザー工業(TAKERU)　☎052(824)2493

★大航海時代

中世を舞台に，7つの海を航海し最高爵位を目指すというシミュレーション。地中海からスター

FAR SIDE MOON

▶ややゆったりすぎる感じは隠せないが，ソフト開発etc.の流れがいい（6月号のP.37の『最後にひと言』には，共感した）。

群馬県・斉木　喜浩(16)

▶いろいろな戦略が立てられるところがいい。

北海道・河野　健司(15)

▶あのグラフィック，あの値段，あのソフトハウス，文句なしですハイ。

京都府・高　宏行(15)

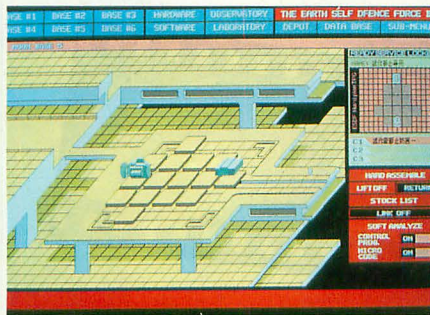
▶作る楽しさを味わえる。

東京都　柴田　和久(17)

▶最近リアルタイム中毒のため。

埼玉県・筒井　圭一郎(17)

このゲームも純粋に楽しめるシミュレーションだったようで，例の件など気にもせずに皆さん遊んでくれているようです。なかなか“ものを作る”“頭を使う”といった点で好評でした。グラフィックもきれいでしたし，人気が高いのもうなずけますね。

X68000用　5″2HD版3枚組　9,500円
アートディンク　☎0474(77)7541

サーク

▶移植に手を抜いてない(みたい)だから。

大分県・佐々木　亮(14)

▶PC-8801のときは，4MHzでしかできなかったから。OPMBには笑えた。POINTゲームは面白い。

千葉県・後藤　勝(21)

▶全体的に動きがいいしきれい。

神奈川県・池本　誠(19)

▶音楽，グラフィック，シナリオ，難易度がよい。

大阪府・平野　晃男(20)

▶イースシリーズと似ないための努力がヒシヒシと伝わってくるから！

神奈川県・石井　直樹(16)

▶最後のボスが強い，けど面白い。これは買いだ。

静岡県・岡田　拓也(17)

比較的最近に発売されたゲームなのに，けっこうたくさんのハガキがきました。テンポよくプレイできるのが要因でしょう。そう難しいクエストもなく，とっつきやすいゲームといえます。

X68000用　5″2HD版4枚組　8,800円
マイクロキャビン　☎0593(51)6482

三国志II

▶他国との外交関係など，いろいろな要素があって，とにかく面白い！(光栄さんのソフトの中ではいちばんいい)

山形県・水口　昌郁(21)

▶武力，知力ともにIよりも史実に近くなった。

滋賀県・田中　哲(17)

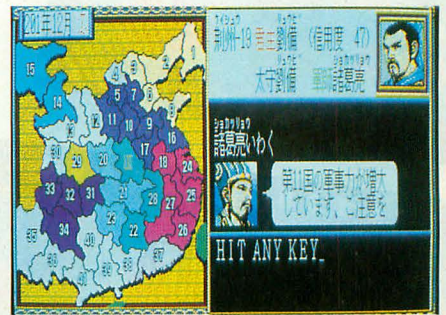
▶三国志IIは確かに難易度が低くなった。また，計略で忠誠度100の武将の引き抜きもできるようになったのはよい。ただ，200/400ライン兼用のためか漢字VRAMの使用を止めたことで反応速度が遅くなったのはおしい。

東京都・中北　康夫(23)

▶前作の欠点がだいぶ改善されているし，MUSIC以外はよくできている。

岐阜県・西松　博章(19)

前作と比べてよくなったという方が多かったですね。かけひきや一騎討ちがよいという方も何人かいました。さすがはシミュレーションの光栄といったところでしょうか。

X1turbo用　5″2D版3枚組　14,800円
光栄　☎044(61)6861

トし，数々の港で仲間を集めつつ，貿易や戦闘，航路開拓などをして進めていく。

X1turbo用　5″2HD版　9,800円
光栄　☎044(61)6861

新作情報

★ウルティマV

アメリカの大ヒット作，ウルティマの最新作がいよいよ登場。

ブリティッシュ国王が地下大陸に消えてしまって以来，ブリタニアの国民は圧制に苦しめられていた。その根源は心の影の化身，シャドー・ロード。再び聖者アバターはブリタニアに降り立ち，人間の内なる正義と悪の葛藤の旅に出る。

X68000用　5″2HD版2枚組　9,800円
ポニーキャニオン　☎03(221)3131

★ユニオン

X68000に新規参入のポニーテールソフト/STUDIOオフサイドによるパズルゲームだ。画面に置かれた牌同士を縦横自由に合体させ，巨大な牌にしてしまうのが目的。キレイに書き込まれた牌は，麻雀・昆虫・果物・動物に自由にコンフィギュレーションできるから，皆さんの好みに合わせてどうぞ。

X68000用　5″2HD版　7,800円
STUDIOオフサイド　☎0722(86)8050

★REINFORCER

ザインの今度の新作はトップビュータイプの8方向スクロールシューティング。敵工場などのステージを，3種類の武器を携えて駆けまわる。3種類のシナリオにしたがって主人公も変化し，各ステージ終了後など，ストーリーの分岐点ではビジュアルシーンによるストーリー進行が用意されている。

X68000用　5″2HD版　価格未定
ザイン・ソフト　☎0794(31)7453

★Gツール

ザイン・ソフトのグラフィックツール。最高512×512ドットまでのスプライト，BG，グラフィックキャラクターのデータを作成することができる。アイコンの編集やマウス両ボタンの機能の設定もできるので，効率的な作業が行えるぞ。

X68000用　5″2HD版　価格未定
ザイン・ソフト　☎0794(31)7453

★実践ビリヤード

PC-9801版「実践ビリヤード」がパワーアップしてX68000に登場する。プレイできる競技は四つ玉・ナインボール・ローテーション，そして最近注目のスヌーカーの4種類。練習モードや予想軌跡の表示もできるので，初心者にもとっつきやすい1本といえそうだ。

X68000用　5″2HD版　9,800円
パック・イン・ビデオ　☎03(5565)8732

★プロミストランド

ポピュラスのシナリオデータ集が発売になる。といっても，単なるマップ集ではないぞ。思考ルーチンはもちろん，キャラクターや舞台まで一新されている。宇宙基地，フランス革命，LEGO，西部劇，コンピュータメーカー(あぶねー)，それと日本版だけのオリジナル，「江戸時代編」など。また違った気分の対戦が楽しめそう。ポピュラスファン必携だ。

X68000用　5″2HD版　4,800円
イマジニア　☎03(343)8911

★C-TRACE ver.3

レイトレーシングソフト「C-TRACE」がバージョンアップして登場。ボクセル分割により計算時間を短縮し，メモリ制限をなくして多量のデータのマッピングを可能にした。さらに，キーフレームソフトの追加やツリー構造の採用によって，モデリング，アニメーション作業の効率化が図られている。

X68000用　5″2HD版　98,000円
キャスト　☎03(705)1065

Oh!X Graphic Gallery

DōGA・CGアニメーション講座

寺田の教育的指導

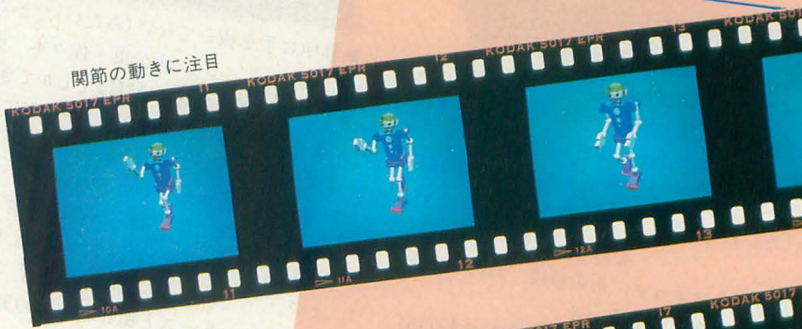
今月の作品は，本職の日本画家という東京都の森山知己さん制作の“走る「MASAYA」”と“メリーゴーランド（仮称）”，それと私こと寺田の作ったプログラム「BOMB.X」とその応用です（本誌103ページ）。

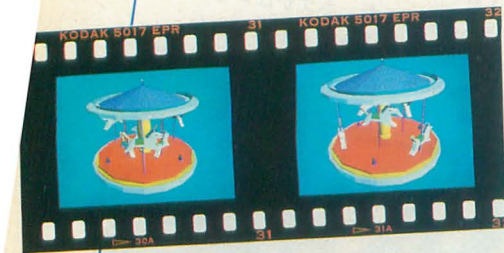
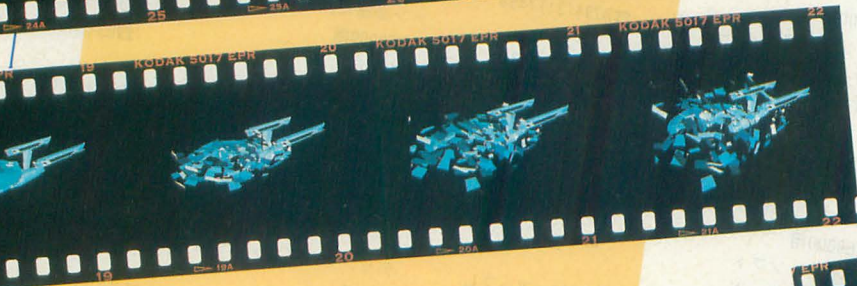
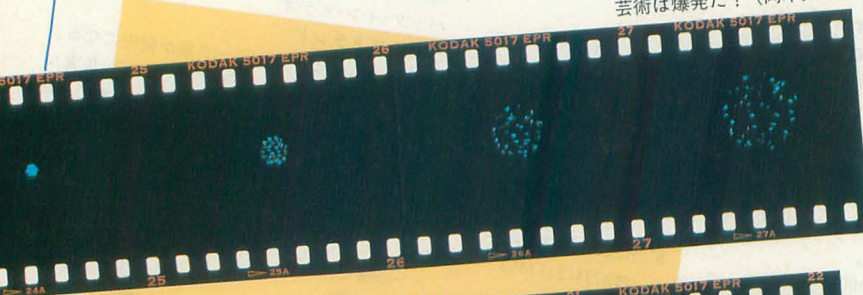
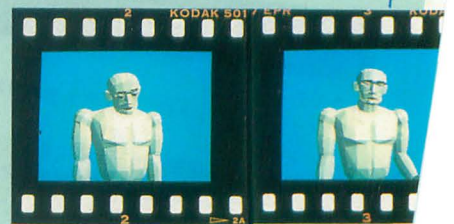
まず走る「MASAYA」は，人間の走っていくさまをCG上で再現したもの。関節の動きなどがとても忠実に表されているので，勉強になる作品です。

メリーゴーランドは，色づかいや雰囲気などメルヘンチックに表現されていて思わずほのぼのとしてしまう作品です。さすがは本職の画家といったところでしょう。

「BOMB.X」は物体の爆発を題材にしたものです。応用作品も一緒に掲載しておきますので参考にしてください。

最後に「人体モデル」の画像データを示しておきます。これも人の動きを表したものです。リアルな感じが出ていると思いますのでじっくり観賞してください。

関節の動きに注目

とても自然にフレームアウトしていく

メリーゴーランドの雰囲気がとてもよくでている

芸術は爆発だ！（岡本太郎）

よくできた人体モデルが……

リアルに動くとさらに気持ち悪い……

●特集　マシン語への第一歩

ASSEMBLER PROGRAMMING

「マシン語」という響きは忘れられようとしている。「マシン語」は本来「アセンブリ言語」とは違う意味あいを持っていたはずだ。整った環境，高度なシステムサービス。68000では「マシン語＝暴走」という図式すら成り立たなくなった。それでもマシン語という響きが求められている。
マシン語をやっている人とそうでない人では歴然と違う。マシン語をやった人はとにかく強い。なんにでも強い。知識とか経験以前にとにかく強い。
8ビットではもうマシン語は必須科目といえる。68000の場合，これからの開発の主力は間違いなくC言語になるとしてもマシン語の知識は有用なものとなるはずなのだ。

CONTENTS

ぜんまいちゃん再び

Yoshida Kouichi
吉田 幸一

吉田幸一氏の復活とともに，あのK君とぜんまいちゃんが帰ってきました。今回はぜんまいちゃんのマシン語に挑戦です。どうすればマシン語とつきあえるのか？　あれ，「ぜんまいちゃん」ってインタプリタの名前だっけ……。

元来，コンピュータというものは素人さんが触るようになるとは夢にも思わず，開発され使われてきたものである。しかし，時代は変わるもので，なんの間違いか素人さんがパソコンとやらを使うようになってきた。さあ，大変。いままではパソコンに触るのはコンピュータに興味のある人たち，大きなコンピュータは専門の技術者がいて，素人さんが触るのは技術者さんたちが作ったソフトやハード（銀行のATMとか）だったから，テキトーに作ってればよかったのだ。

そこで，素人さんが中を知らなくてもいいような新しいコンピュータが登場した。日本では“ファミコン”と“ポータブルワープロ”，アメリカでは“マッキントッシュ”だ。そんでもって，世のたいていのパソコンは素人さんが触るなんて考えてなかった設計だったのだが，世界中で素人さんが使えるパソコンやソフトを！　てな感じで盛り上がっていったのであった。

ちょっと待った，と，誰もいわないから私がいおう。コンピュータって，そんな，着せ替え人形みたいにソフトを取っ換え引っ換えして便利になればいいものだったっけ。ゲームがたくさんあって，ソフトが多ければいいものだっけ？

違うのである。パソコンをただの着せ替え人形にしてしまったのは，コンピュータを特権階級のものにしておきたかったフォン・ノイマンの亡霊であり，国家の陰謀である。これからネットワークでいろんなものがデジタルでつながるようになると，その元であり端末であるコンピュータが重要なポイントになる。それをブラックボックスにしておきたい人がいるのだ。権力であり，企業である。彼らはコンピュータが一般市民にとってブラックボックスであるのをいいことに，マインドコントロールをしようとしているのだ！　立ち上がれ！　ハッカーたち。都合の悪い情報を隠そうとする権力と戦うのだ！　はぁ，はぁ。

パソコンは個人が個人で個人を守る武器である。我々は権力に屈してはならないのだ。閉じた世界は銀行や商社のデカコンネットワークだけで十分。はぁ，はぁ。

パソコンの3つの顔

おっと，話が全然関係のない方向へ進んでいるぞ。元に戻そう。

パソコンはとにかくとんでもない代物である。そのパソコンには3つの顔がある。

ひとつは実務道具，事務機器としての顔だ。たとえばパソコンの能力を利用して，金儲けの手助けをしたり，面倒臭い虚礼的な手紙を書くのが面倒だからとひとつの文章を使い回ししたり，技術者が図面描いたりするときの顔だと思えばいい。まあ，たいていは従来あったものをよりよく管理しようというシミュレート的な色彩が濃かったりする。

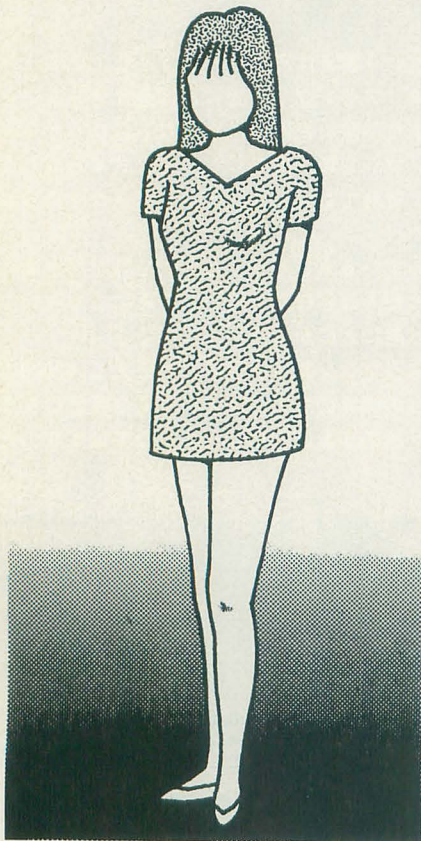
もうひとつはいま流行っているメディアとしての顔だ。たとえば電脳倶楽部であり，ゲームであり，あなたが書いたグラフィックのデータであり，モデムを使って遠隔地とメッセージのやりとりをしたりするときの顔だ。ここで重要なのが，パソコンはテレビや本などの既成のメディアと違って情報の発信と受信の両方ができること，それからデジタルデータにできる情報なら絵だろうが文だろうが音だろうがなんでも扱えることだ。

さて，3つめの顔がコンピュータそのものとしての顔である。人間に対して挑戦的なほど論理的に動くコンピュータ。幾多の人間がその魅力に捕らえられたことだろう。パソコンはコンピュータの面白さに取り憑かれた人々の手によって発展したといってもいい。日本ではシャープが，道具やメディアとしてはイマイチだけど，コンピュータとしてはとても面白いパソコンを作り続けていた。ねえ，MZ君たち。

このコンピュータとしての顔を知らずして，パソコンの本当の面白さを知ることはできないし，パソコンを値段分使いこなすことは難しい。世間ではメディアとしてのコンピュータが脚光を浴びているが，その陰でコンピュータそのものを楽しむ人がいなければなにも生まれないのだ。

もっとも重要なのはメディアとしてのパソコンとコンピュータそのものとしてのパソコンの2つだ。パソコンをコンピュータとしてしか楽しめないような視野の狭い人はいらないよ，と釘をさしておこう。

と，いうわけで，私はコンピュータそのものを楽しむためのパソコン入門をここに展開することにしたのである。脆くてコピーしやすくて盗みやすいデジタルデータを権力のものとしないためにも，コンピュータを知る人たちを絶やしてはいけないのだ。

おっと，このままでいくとどんどん話が大仰になっていきそうなので，バキっと気持ちを入れ替えよう。

へろへろへろ。ふにゃふにゃふにゃ。はい，OK。

では，なんというか，何年も前にたった1回登場させただけなのに，いまだに復活を望む声がちらほらする“ぜんまいちゃん”でもいってみようか。あーあ，なんか，書くほうとしては恥ずかしい。知らない人も多いだろうしなあ。別に知らなくてもいいんだけど。下手に書いてしまって，あーあ，きれいな思い出にしときたかった，てなことになるのもなあ。うーん。うーん。

決心。だいたい，きれいな思い出なんて食べるのがもったいなくて腐ったメロンのようなものである。そんなものに頼るやつがいけないのだ。

てなもんで，ぜんまいちゃんとK君です。上の文章から察するに，K君がぜんまいちゃんを通じてコンピュータとはなにか，どーすれば動くかという話へと発展していくのではないか，と予想されます。

K君とぜんまいちゃんの話

月日のたつのは早いもので，K君はぜんまいちゃんユーザーになってもう3年になります。愛と希望のぜんまいちゃん。手垢にまみれたぜんまいちゃん。でも，女をくどくには金でも顔でも心でもなく，マメさが一番（第1法則）だと知っているK君は，ぜんまいちゃんにもマメです。手垢や汚れが気になり出すと，ぜんまいちゃんクリーナーで拭き拭きし，それでもだめならちゃんとメンテナンスセンターへ行って，皮膚の張り替えをします。もはやぜんまいちゃんはK君にとって，ただの馬鹿チンアンドロイドではないのです。ほら，猫嫌いの人でも3日も飼えば情が移るというではないですか。で，異性をくどくには情を移せば勝ち攻撃（第2法則）を図らずもぜんまいちゃんからくらってしまったのでした。

そんなある日，K君はぜんまいちゃんのライバル機種「全自動“私を好きにして”ちゃん」(筆者注：おっと，これでは吉田戦車のパクリそのものではないか。吉田戦車さん，ごめんなさい）を買ったM君に大自慢されたのでした。

「ぜんまいちゃんにお茶とお華ができるか。全自動“私を好きにして”ちゃんは，6000枚あるICカードでなんでもやってくれるんだぜ」

おっと，“私を好きにして”ちゃんは全自動というだけあって，ソフトさえ交換すればなんでもできるのでした。

K君はといえばぜんまいちゃんをぜんまいちゃんインタプリタという日本語ライクな言語で動かしていたので，複雑な動作だけでなく，高速な動作も苦手なのでした。しかも，微妙な動作を要求すると，フィードバックコマンドが速度についていかず，行ったり来たりの“中風”状態になってしまうのです。

K君はハックおじさん(1987年5月号参照……するほどのこともないけど）を頼ることにしました。フラットなディスプレイに移ったおじさんの顔。K君は受話器に向かって，話の尾鰭をつけて切々と語ります。

ハックおじさんはいいます。

「それはぜんまいちゃんが悪いんじゃない。ぜんまいちゃんのCPUもハードウェアもお前がいったことができるくらいの性能はあるぞ。まあ，ぜんまいちゃんのハードを勉強でもして，直接動かしてみることだな。機械を動かすには機械語が一番。おっと，愛人が来たので失礼するよ」

なんてアバウトな説明。

K君は「どーせ愛人ったってレプリカントだろうに」などと悪態をつきながら，うーん，とぜんまいちゃんを見つめます。確かに，ぜんまいちゃんが動くだけで感動した3年前とは違い，ただ日本語風言語で動くだけのぜんまいちゃんにはちょっともの足りなさを感じています。そのうえ，最近はぜんまいちゃんがどうして“日本語の命令”で動くのか，疑問に思ってもいたのです。

ぜんまいちゃんの頭脳はモトトール社のハイリスクプロセッサ，強欲1号です。超単純なコマンド体系といい加減な割り込み処理が特徴。

ぜんまいちゃんはデジタルなコンピュータですから0か1かの信号しか受け付けません。そのくらいはK君も知ってます。

「要は0か1か区別がつけばいいんだな」

とりあえず，あいさつをしてやろうと，JISコード表を調べ，「おはよう」を16進の「82A882CD82E682A4」にして，「1000001010101000100000101100100000101110011010000001010100100」と2進数を書いてぜんまいちゃんに読ませました。もちろん，ぜんまいちゃんはそんなもの受け付けません。

1と0の並びを受け入れるのはあくまでもCPUの強欲1号であって，ぜんまいちゃんのCPUとK君のあいだには，CCDパターン認識カメラやらそれを解釈する信号プロセッサやらさらにそのデータを翻訳するソフトウェアなんかがゴテゴタとあって，それではじめてぜんまいちゃんはなにかをするのです。この場合は，1000001010101000100000101100100000101110011010000001010100100がそのままCPUに到達したのではなく，コマンド解釈・実行プログラムの入力チェックにひっかかってしまって，あえなく玉砕したのでした。ま，当たり前ですけど。

「うん，そうくると思ったぜ」

K君はめげません。そんなとき，ハックおじさんから電話がかかってきまして，「どーせお前のことだから，指の1本も動かせんだろう」

まさにそのとおりで。

「でもおじさん，機械って1と0しか使えないんでしょ？　もしかして1011001100って，やんなきゃいけないの？」

「いや，それでもまだ考えが甘い。だって，機械にとってみればいまのだって，1という字と0という字があるだけじゃない

前回までのあらすじ

どじでのろまなK君は高校生の分際で“ぜんまいちゃん”という女性型アンドロイドを買ったのであった。うーん，金持ち。そんでもって，若者らしく“機械相手のエッチ”でもするのかと思いきや，なんと，ぜんまいちゃんに搭載されている言語でぜんまいちゃんを動かそうとするのである。おお，女と金にしか興味のないやつに読ませたいくらいだ。

で，「ぜんまいちゃんインタプリタV1.0」を使い，いろいろとぜんまいちゃんを動かすうち，見た目は人と変わらなくてもコンピュータはコンピュータであることと，“人間がいかに抽象的でいいかげんで暗黙の了解と一般常識に頼って生きているかを知る”（1987年4月号）のであった。

「ぜんまいちゃん」というのは富士山麓地下にある村正研究所で開発された女性型アンドロイド（全快1号）用の言語だが，全快1号自体の愛称として使われることも多い。ちなみに全快1号の身長は160cm，美人というより可愛いというカンジの女の子仕様。フィリップ・K・ディック風にいうと，レプリカントかシミュラクラ……。得意技は「ぷーっ，間違ってまーす」攻撃だ。

全快1号を開発した村正研究所の所長は自称“祝一平の永遠のライバル”呪一平である。「ぜんまいちゃんインタプリタV1.0」は正宗研究所の開発による完全日本語仕様のインタプリタ言語だ。詳しくはOh!MZ1987年4月号を参照してほしい。もしかしたら，全快1号はもとより，元ネタになった「満開1号」「満開2号」を知らない人も多いのでは……という気もするので可能ならばOh!MZ1987年3月号も参照したほうがいいかもしれない。

*　*　*

当時私はMZ-2500のユーカラK2というワープロで原稿を書いていた。うーん，大昔のことのようだ。いまはX68000でHyperwordだもんな。

か」

ハックおじさんはニタニタと“笑ゥせぇるすまん”笑いを見せます。

「じゃあ，どうすんの？」

「そりゃあ，だな，ぜんまいちゃんの背中を開けると，スイッチがずらっと並んだパネルがあるから，それをひとつずつオンにしたりオフにしたりするんだよ。そうすれば，直接電気でぜんまいちゃんに伝わるぞ」

K君はア然としたまま石になりました。おじさんはさらに，「で，だな，パチパチスイッチを入れて，入力スイッチをペンッと入れると，パチパチしたところを電気が流れて，ぜんまいちゃんのメモリに流れるわけだな。で，作ったプログラムを流し込んだら，実行させればいいというわけだ」と追い打ち。K君はそんな面倒なことをしなければならないとは露知らず，思わず「全自動“私を好きにして”ちゃん」に買い換えようかと思ったのでした。プログラミングはできなくても，何千枚のICカードがありますから。

でも，よく考えたK君。考えなくてもわかりますが，ぜんまいちゃんの背中にパネルがあるなんてどのマニュアルにもネタ本にも書いてありません。

「ところでおじさん，それ，本当？」

「わしが本当のことをいったことがあったかね」

そして“笑ゥせぇるすまん”笑い。

「じゃあ，どーするんだよぅ」

「ちゃんと機械語でプログラム作る人のためのソフトウェアがあるから，安心しなさい」

K君は「どーしてぜんまいちゃんが直接理解できる機械語なのに，専用のソフトがいるのだろう」と思いましたが，少しでも楽できるのは（パネルのスイッチよりは）いいわけで，「頂戴」といいました。おじさんは，電話回線（筆者注：この頃はデジタル回線になってたりする。楽観的すぎるけど）でそのソフトを転送してくれました。そのソフトで作ったプログラムはぜんまいちゃんインタプリタとぜんまいちゃんの仲立ちをしているぜんまいちゃんOSの上で走ります。

「ついでに，スクリーン上で動作確認ができるぜんまいちゃんエミュレータもあるけど，ほしいか？」

K君は，プログラムよりぜんまいちゃんと遊ぶことが好きだったので，「いらない」といいました。

ぜんまいちゃんを転ばせること

K君が一度やってみたかったことのひとつに，ぜんまいちゃんを転ばせるというのがあります。ぜんまいちゃんには世界一優秀なインテリジェントバランサーが搭載されており，それが働いてる限り，どんな姿勢をさせてもバランスが崩れることはなく，また，バランスが崩れるような姿勢は普通に使っている限り無理なのでした。

K君が調べてみると，ソフトウェアでバランサーのON/OFFは可能なようなので，それをやってみようと思ったのです。

K君がまず調べたのはぜんまいちゃんのCPU，強欲１号のコマンド。なんか，アセンブラとかいうのを使うと０と１の並びを打たなくてもいいそうで，なおかつ０と１の並びでできることは全部やらせることができるそうで，なによりです。

で，強欲１号のコマンドですが，３種類しかありません。演算と移動と分岐です。

「演算・移動・分岐。演算・移動・分岐」K君は口の中で何度もつぶやいてみましたが，なにがなんなのか要領がつかめません。どうすれば「気をつけ」とか「お座り」といった命令が“演算と移動と分岐”になるのか，そのあいだを埋めるものがサッパリなわけです。

あくる日，『コンピュータの基礎』という退屈そうな本を借りてきてやっと次のことがわかりました。

演算というのは，論理演算と数値演算があって，普通は数値演算のことを計算，というらしいこと。

移動というのはぜんまいちゃんが移動することではなくて，メモリにあるデータやアドレスを移動させたりコピーしたりすること。

分岐はメモリ上のいま実行している場所を指すデータの内容を書き換えること。さらに，条件分岐や割り込みによる分岐やユーザー定義命令による分岐があるらしいこと。

K君の頭はクラクラ。初めてぜんまいちゃんインタプリタを動かしたときはマニュアルを見ながら日本語の命令をいってやればなんとかなったのですが，機械語となると，そうもいかないらしいのです。

そのうえ，これだけではどう考えてもぜんまいちゃんの指１本たりとも動かせる気がしません。

「うーん，まだなんか秘密があるはずだ」

すっくと立ち上がってぜんまいちゃんに向かって叫ぶK君。ぜんまいちゃんはスタンバイモードでエネルギー消費を抑えているので目を閉じたままうんともすんともいいません。

ずるずるマニュアルを読んでますと，ROMというのがあって，それに“ぜんまいちゃんギアボックス”という名前がついているのを発見しました。これは怪しい。ROM。工場出荷時にはもうデータが詰まっていて書き換えられないメモリというのが怪しそうです。K君は「どーしてROMの反対がRAMなんだろう。ROMがリード・オンリー・メモリなら，もうひとつはRWM，つまりリード＆ライト・メモリのはずじゃないか」などと思いましたが，これ以上悩みを増やしてはいけないという理性の声によって留まったのでした。

で，ギアボックスと呼ばれるROMの中にはどうやらぜんまいちゃんを動かすためのプログラムが何千個も入っているのでした。

サンプルプログラム

```
 1:  エントリ：
 2:          ユーザー分岐    初期化
 3:          ユーザー分岐    気をつけ
 4:          移動            アドレス（$ED001000），データ（$FFFF）
 5:          移動            アドレス（足上げのデータ），アドレスレジスタ１
 6:  ループ：ユーザー分岐    右足，アドレスレジスタ１
 7:          ユーザー分岐    バランサー角度読み込み，データレジスタ１
 8:          引き算          60，値（データレジスタ１）
 9:          条件分岐        式の値が零，ループ
10:
11:  足上げのデータ：                     ＊ボディコン用データセット
12:          データセット    $08          ＊ボディコンデータ受信モード
13:          データセット    $21          ＊身体部分ＩＤ
14:          データセット    $48          ＊動作モード
15:          データセット    $00          ＊慣性モーメント
16:          データセット    $CE          ＊コントローラ１用ベクトル
17:          データセット    $2A          ＊コントローラ２用ベクトル
18:          データセット    $99          ＊対ショック対閃光防御
19:                          ：
```

でもK君は納得できません。だって，プログラムが何千個あろうが何万個あろうが，使っているのは演算と移動と分岐だけなはずです。

「これはまだ秘密があるに違いない」

そんなとき，ふと目に入ったぜんまいちゃんのカタログ。よく見ると，最新型ボディコントロールユニット搭載，とか，音声認識ユニットの高性能DSPとか書いてあるではないですか。そうです。CPUだけではないのです。

「うーん，問題なのは，CPUがどうやってボディコントロールユニット（略してボディコン）やら音声合成・認識ユニットやらに命令を出しているかだ」

パラパラパラパラ。K君，腕を伸ばして……。

「よう，こんな夜中にどうした。え？　まだあきらめてないのか。そーか，そーか。え？　そりゃあ，CPUがボディコンにボディコン用の命令やデータを渡しているだけだよ。ボディコンはそれを受け取って手足や腰なんかのコントローラに命令を送って，そいつらが実際に動かしてるわけ。どーすればできるか？　メモリの勉強をしたか？　アドレスがなにかわかってるか？　なんなら，教育用のコンピュータでも貸してやろうか」

「やだやだ，ぜんまいちゃんでなきゃやだ」

「んなら，簡単に喋ってやるから，あとでちゃんと勉強するように」

「へいへい」

「アドレスの一部がメモリの代わりにボディコンやら音声合成やらICカードコントローラやらとの入出力用に割り当てられ

ていてだな，ボディコンだったら，ボディコン用のアドレスにボディコン用のコマンドやらデータやらを書いてやればいいわけだ」

K君はまたひとつお利口さんになった気がしました。これなら，演算と移動と分岐だけでぜんまいちゃんを転ばせられそう。直接ハードウェアを操作できるボディコン用の命令を，移動命令を使って，ボディコンが読めるところへ動かしてやればいいのですから。

ボディコンにバランサーをOFFにする命令を送って，右足だけを少しずつ上げていけば，すぐ転ぶはずだ，と，K君はアルゴリズムを練ります。それなら簡単じゃん，と。

「よし，まずは初期化だな。最初に気をつけをして立たせておかないと駄目だからな」

で，気をつけはインタプリタにあるコマンドなので，絶対ギアボックスの中に用意してあると思って調べるK君。

ありましたありました。姿勢の初期化。“ユーザー分岐　気をつけ”。これをアセンブラにかけると“1010001110110011”というコードになって，ギアボックスの気をつけをさせるプログラムのあるところへ分岐して，そこでボディコンに気をつけをさせるための命令がダダダーっと流れるのです。

そのあいだ，CPUは気をつけが終わるのを待って，終わったという信号を受け取ったら，

「気をつけをしたら，バランサーのOFFだな。これはギアボックスにはないだろうから，直接ボディコンにコマンドを出さなきゃ。えっと，バランサー制御のアドレスは16進でED001000だから，そのアドレスに，えーと，えーと，FFFFをセットしてやればよし，と。セットするのは移動コマンドだな。と」

“移動　アドレス（ED001000），データ（FFFF）”

「次は右足だな。“ユーザー分岐　右足　アドレスレジスタ1”と」

これはアドレスレジスタ1に入っているアドレスの指しているメモリのデータをボディコンの右足制御コマンドのパラメータとして渡すといったものです。この場合のユーザー分岐のユーザーというのは，CPUメーカーにとってのユーザーということで，つまり，ぜんまいちゃんを作った村正研究所のこと。村正研究所が右足ときたら右足だぜ，ということにしたわけです。

「それには，右足を動かす前に，アドレスレジスタ1にアドレスをセットしなきゃならないし，その前に，そのアドレスに右足を動かすデータをセットしとかなきゃ，と。そんでもって，右足を少しずつ動かしたいから，ループさせて，と。ループ終了の条件も入れなきゃ。倒れても足を上げ続けるなんて醜いからな。壊れるかもしれないし。

「どーしようかな。えっと，バランサーは姿勢制御をしないだけで，現在姿勢のデータは取っているはずだから，そのデータを読んで，60度くらいまで倒れたら実行を止めさせればいいだろう。それは，と，ギアボックスにあったから，大丈夫だな」

そんなこんなで，このあとぜんまいちゃんが暴走して100mを9秒で走っていってしまったり，うんともすんともいわなくなって鼻のリセットボタンを押したりしながら，なんとか図のようなプログラムができたのでした。

ぜんまいちゃんのOSは音声オペレーションですから，プログラムの名前をいってやれば実行してくれます。

「お転び！」

これがK君のつけたプログラム名。ぜん

吉田の謎

どーでもいい話だが，どーして私が約2年の間休んでいたのかというと，決して高橋留美子ファンに殺されるのが怖くて逃亡していたわけでもX68000を見捨てて98ユーザーになったわけでもない。某金融機関系ソフトハウスという固い職場でIBMちゃんの固いマシンの相手をしてたからである。

そして，大方の「そんなところ，お前に勤まるわけがない。3カ月で辞めるほうに1万円！」という声にも負けず，めでたく2年ももったのであった。

当人は円満退職だったと思っているが，そうでない人もいたらしい。どーして辞めたのかというと，それは祝一平氏に「あやしい」といわしめたくらい謎である。ついでに，いま，なにを本職にしているかも謎である。

で，今後どーなるかも，ぜんまいちゃんがどーなるかも謎である。ま，いまの日本のアバウトで閉じた平和ほど謎なものはないけどね。

まいちゃんは一瞬の内に右足を上げたかと思うと，ドタン，と，無事，K君を下敷きにして転んだのでした。K君，うまく動いて嬉しいやら痛いやら。

K君は腰のあたりを打ったせいか力ない声で「気をつけ！」と，コマンドをいいます。すると，ぜんまいちゃんはすっくと立ち上がります。

「はあはあ，この野郎，わざわざ俺の上に転ばなくても」と，ぜんまいちゃんに聞こえないような小さな声で噛いて立ち上がり，ぜんまいちゃんを「こいつぅ」とこづきました。

すると，軽く当たっただけにもかかわらず，ぜんまいちゃんはドタンと倒れたではないですか。階下から「さっきからドタンドタンとうるさいわよ！」と苦情もきてしまい，K君はあせります。

「そうか，バランサーをONに戻し忘れていた」

そして，ガチャン。

「おーい，K。3日も学校へ来ないでなにやってたんだぁ」

と，ノックもせずに入ってくる悪友たち。代わる代わるぜんまいちゃんを撫で回します。

「お転び！」

ぜんまいちゃんはドタンとその悪友のひとりの上に倒れました。

「ぜんまいちゃんを転ばせるのに3日かかったんだ」

と，鼻の下を指でこすって，K君。

「こんなことで3日も休んでたのか，バカなやつ！」

そういって彼の友達はぜんまいちゃんの下敷きになってげらげらと笑っているのでした。

自分で作らなくとも

K君は謙虚にも人の作ったプログラムを読もう，と，決意します。転ばせるのに3日かかっていたら，ピアノを弾かせるには何年かかるかわかったものではありません。バランサーをOFFにすればぜんまいちゃんはかなりの速度で動くことができるので，転ぶ心配のない上半身だけ使う技は超絶技巧が発揮できるのです。

そこで，ネットワークを覗いてみますと，ぜんまいちゃん用の機械語のプログラムがたくさんあるではないですか。K君は腕だけを使うプログラムを探しました。と，見ると，“シャープペン回し”というプログラムを発見。

作者のメッセージを見ますと，「親指と人指し指のあいだでシャープペンシルを回すプログラムです。何十年も前から受け継がれた技をぜんまいちゃんに引き継いでみました」となっています。K君はその技を知らなかったので（だいたいシャープペンシルなんてあまり使いません），ダウンロードしてみました。

ダウンロードした中にはアセンブラのプログラム，実行プログラム，マニュアルがついてます。マニュアルによると，ぜんまいちゃんを座らせ，両手にシャープペンシルを持たせる必要があります。ちゃんと正しい握り方で。そして，回せ，というと，ぜんまいちゃんが器用に両手で上手にシャープペンシルを回すではないですか。くるくる。くるくる。シャープペンシルが指の上で回転しているのです。

「おお，なんだ，この技は」

ぜんまいちゃんが微笑顔で（ちゃんと微笑むルーチンがプログラムに入っていた）くるくるシャープペンシルを回す様は異様です。

K君は自分でもやってみたくなったので，じっくりぜんまいちゃんの指を観察します。マニュアルには「速く」と「ゆっくり」と「やめれ」の3つのコマンドを受け付けるとあったので，とりあえず，

「ゆっくり」

「ゆっくり」

2回くらいで，なんとか，指の動きがわかる速度になりました。

「ぜんまいちゃんって，凄い。こんなこともできるんだ」

できてどうするんだという話もありますが，K君はシャープペンシルを持って，ぜんまいちゃんの横で必死に練習するのでした。何度もシャープペンシルを落として，中の芯をぼろぼろにしながら。

やっとぜんまいちゃんと一緒に回せるようになった頃。

そして，ガチャン。

「おい，K，今日はなんで休んだんだ」

と，悪友はぜんまいちゃんを見ると，

「すげえ。これ，お前がやらせたのか」

と，K君が一緒に回しているのを横目にいった。K君は，

「いや，ぜんまいちゃんがやってるのを見て俺が練習してるんだ」

「そんなことで休んだのか，やっぱ，バカなやつ」

そういうと悪友はけらけらと笑ったのでした。

K君はというと，おいおい，人のプログラムを見て勉強しようという意気込みはどうしたんだ。くるくる，くるくる。

*　　*　　*

と，いうわけであった。今回は高水準言語から機械語へ降りていくという従来見られたステップとは逆に，0と1からアセンブラへというもっと下から上っていく過程を書きたかったのだが，意図したところがわかってもらえただろうか。

機械語なんて言語だと思うからいけないのであって，あんなのはただの記号の羅列である。それをひとつずつ動作を考えながら積み重ねていく。これがはじめの一歩。

アセンブラでバリバリプログラムを作れ，なんて私はいわない。ただ，一度でもアセンブラを経験し，コンピュータがどう動いているか，どういうことならできてどういうことはできないかを感覚的に知ることはすすめる。それは大変だけれども有用なことだ。どーせ，「ソフトはハードを壊せない」のだから，安心して暴走させればいい。そして，頭を使って悩めば少しは謙虚な気持ちになれるだろう。

さらに重要なのは，パソコンを使うのに必要なのは知識ではなく，知恵と感性だ，ということである。だから，ぜんまいちゃんのように完全架空の機械のほうが，知識に頼らないだけ，感覚を掴むにはいいかもしれない。

私はもう年である。ぜんまいちゃんもこれで終わりにしたいので，もうぜんまいちゃんがどーしたとかこーしたとかいわないように。では。

参考文献紹介

ラジカルなパソコン入門　岩谷宏著　ちくまライブラリー27　筑摩書房

この著者はマウスに偏見を持っている，PC-9801とIBM-PCしか認めていないような感がある（X68000のことなど知らないのだろう），内容に多少怪しいところがある，パソコンを本文で書いた3つの顔のうちコンピュータそのものとしての顔としか見ていない（すべてその視点から語っている）といった気に入らないところはあるにしろ，内容はまっとうである。MS-DOS入門でもパソコンの歴史入門でもない，希有のパソコン入門書だ。ソフトウェアの著作権を認めない（それだけならただの馬鹿だが，この人は著作権自体認めないそうなので，それはそれで筋が通っている）という態度も面白い。著者の個性がはっきり出過ぎているので反発などもあろうが，初心者にこびない良書だと見た。

戦え！　軍人くん　吉田戦車著　スコラ社

いうまでもないです。ちなみに，私は“友達の家へ泊まるっていってきたの”ちゃんが好きです。

そのほか，P.K.ディックのさまざまな小説

マシン語ってなあに？

Mounai Toshiyuki
毛内 俊行

世の中にはたくさんのCPUがあり，それとほぼ同じだけ違ったマシン語の作法があります。しかし，それらの根底となる部分はどれも同じようなもの。ここではZ80，68000をまとめて，マシン語特有の考え方について解説します。

マシン語ってなんでしょう？　マシン語はコンピュータが唯一理解できる言語です。しかしマシン語は，我々のよく使うBASICやCとは，まったく異なった独特な仕組みや表現を持っており，初めて勉強しようとする人にとって，なかなか理解しにくい言語でもあります。

しかしほかの言語と比べると，プログラムの実行速度が桁違いに速く，サイズも小さくてすむというマシン語の魅力は，一度体験したらもう忘れられません。ここではそんなマシン語の概念を理解して，マシン語とは実際にどんな言語なのかを理解していこうと思います。

コンピュータの仕組み

現在のコンピュータは，ノイマン型コンピュータと呼ばれるもので，ドイツ系アメリカ人(だと思ったんだけど)のフォン・ノイマンという人が提唱した仕組みのものです。ノイマン型のコンピュータの特徴は，実行内容を記述したプログラムをメモリ上に置き，それをCPUが1つひとつ読みながら実行していくところにあります。

たとえば，メモリを大きなマンションだと仮定してみましょう（大胆！)。あなたがそのマンションへやってきました。最初に1号室の扉を開けるとそこに1通のメモが置いてありました。それには，

「3号室の扉を開けなさい」

と，書かれてあったので，あなたは3号室まで歩いて行き扉を開けました。するとそこにも1通のメモがありました。それを読むと，

「10という数を覚えておくように」

と，書いてあります。あなたは忘れないように自分の手帳に「10」と書き込んで，隣の4号室の扉を開けました……。

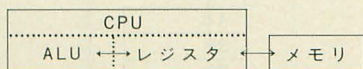



図1　CPUとメモリの関係

さて，これはちょっと大胆な例ですが，コンピュータの中では，いつもこんなことが行われているのです。ここではあなたがCPUの代わりになって，メモリであるマンションの部屋を1つずつ訪ね，そこに置かれているメモを実行していきます。これがCPUとメモリの関係です。

メモリはマンションの部屋番号のように番地（アドレス）があり，各アドレスには00_H～FF_Hの16進数が入ります。この数字1つひとつがCPUに対するメモとなり，このメモの羅列がマシン語のプログラムとなるわけです。

さて，それではCPUの内部をちょっと見てみましょう。CPUは図1に示すように，大きく分けてALUとレジスタ部に分かれます。ALUは演算装置のことで，ここであらゆる計算が行われます。そして，レジスタとは記憶装置のことです。

先ほどのマンションの例を思い出してください。たしか3号室へ行ったとき，10という数を忘れないように手帳にメモしたはずです。レジスタはこのメモのような役割を持っています。そして、メモリとCPUはこのレジスタを介してデータのやりとりを行っているのです。参考までに，Z80のレジスタと68000のレジスタを図2に示しておきます。

仮想マシンBUG

今回は「Z80ユーザーも68000ユーザーも読めるマシン語入門」という，大胆不敵な原稿書きを申し渡されてしまいました。2つ返事で引き受けてしまったものの，どうしていいのかわからなかったので，新しく仮想CPUを作ってしまうことにしました。その名もBUG(この名前に深い意味はありません)。これ以後，マシン語の概念的なところはこのBUGを使って説明していきます。BUGのアセンブリ言語はZ80のそれによく似ています(これは私が68000のアセンブラよりも，Z80のアセンブラのほうが使い慣れているためです）が，レジスタ構成は図3に示すように，ほかのCPUに比べてずっと単純です。

R0～R2は8ビットのデータレジスタ，A0は16ビットのアドレスレジスタだと考えてください。

そのほかに，フラグレジスタ(FLG)，スタックポインタ(SP),プログラムカウンタ(PC)の基本的なレジスタが用意されています。さて，BUGの概要がわかったところで，早速マシン語の命令について説明していきましょう。

データの転送

まずはマシン語のもっとも基本的な命令で，なおかつもっとも頻繁に使われる命令である，データ転送命令について考えてみましょう。データ転送命令というのは，レジスタとメモリのあいだで，データの受け渡しをする命令です。たとえばBASICなどで考えれば，

HENSU＝1

と，いった感じの命令です。BUGではデータ転送命令は，大きく分けて次の3つになります。

レジスタ←　定数
レジスタ←→レジスタ
レジスタ←→メモリ

68000ではこのほかに，

メモリ→メモリ

といった内容の命令もあります。それではひとつずつ説明していきましょう。

・レジスタ←定数

BUGではデータ転送命令はLDを使います。たとえば，R0レジスタに1を代入する命令は，

LD R0,1

と書かれます。ただし，

LD 1,R0

と書くことはできません。これはBASICでR0＝1は実行できるのに1＝R0と書けな

いのと同じです。転送命令では，転送先のレジスタが，必ず転送元よりも前に書かれてなければいけません（68000のアセンブラではこの位置関係が逆になります)。

さて，上の命令と同様に考えて，R2レジスタに50という数を転送する命令を考えてみましょう。そう，簡単ですね。

```
LD R2,50
```

と書けばいいのです。

・レジスタ⟷レジスタ

さて，次はレジスタ同士のデータの転送です。R0レジスタの中に格納されているデータをR1レジスタへ転送してみましょう。つまり，BASICでいうところのR1=R0を実行してやればいいわけです。これも転送命令なので，LD命令を使います。

```
LD R1,R0
```

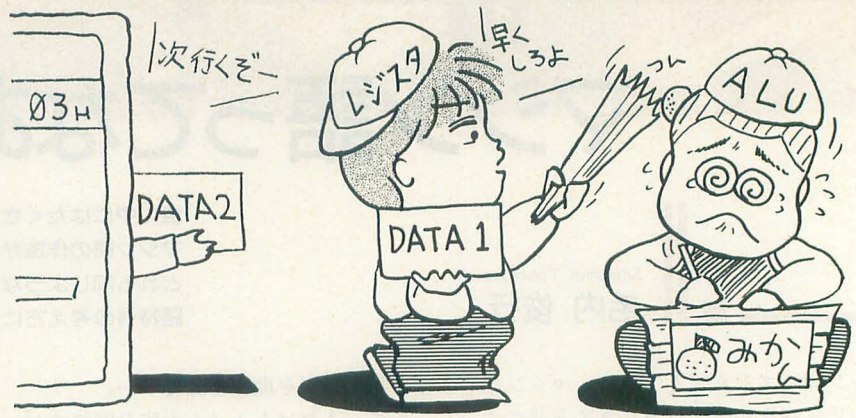
これで，OKです。先ほどの定数を扱ったときと同じように，転送先，転送元の順でレジスタ名を書いてあげればいいのです。なおこのとき，転送といっても転送元の内容は消えてしまうわけではなく，転送後も転送元にデータが残ります。これはすべての転送命令において同じです。

・レジスタ⟷メモリ

さて今度はレジスタとメモリのあいだでデータの転送を行います。今度は先に命令を見てもらいましょう。

```
LD R0,($F000)
```

さて，これはなんでしょう？ 使われているのはLD命令ですから，転送命令に違いありません。するとR0が転送先，($F000)が転送元ということになります。これはR0レジスタに，メモリのアドレス$F000に格納されている数を転送しなさいという命令なのです。このように数字を括弧でくくった場合，この数字で示したメモリ上のアドレスの操作を行うようになっています。

また，メモリアドレスの頭に付いている$は，この数字が16進数であることを示しています。つまり，$F001に格納されているデータをR1レジスタに転送したい場合は，

```
LD R1,($F001)
```

と，なります。

また，メモリ・レジスタ間の転送命令は，逆にレジスタからメモリへデータを送ることもできます。つまり，

```
LD ($8000),R0
```

と書けば，$8000番地にR0レジスタの内容を転送するのです。

例題1　以下のプログラムを見て，次の問いに答えよ

```
LD R0,$FF
LD R1,0
LD ($8000),R0
LD ($8001),R1
LD R1,($8000)
LD R0,($8001)
```

問　上記のプログラムを実行したあとでR0，R1レジスタに格納されている数を答えなさい。

解答

R0=0，R1=$FF

※間違った人は間違いがわかるまでこの先を読んではいけません。

さて，まだデータ転送の話は続きます。BUGにはA0という16ビットのアドレスレジスタが存在します。今度はこれを使ってみましょう。

```
LD A0,$F000
```

これは，A0レジスタに$F000という定数値を代入する命令です。いままで使っていたR0～R2レジスタは8ビットのレジスタだったので，扱えるデータは$00～$FF（0～255）までだったのに対して，A0レジスタは16ビットなので，扱えるデータは$0000～$FFFF（0～65535）までとなりますが，それ以外に変わったところはありません。それではこれはどうでしょうか？

```
LD R0,(A0)
```

転送元が(A0)となっています。つまり，A0レジスタが示すメモリアドレスのデータをR0に転送するという命令です。

つまり，

```
LD A0,$8000
LD R0,(A0)
```

というプログラムと，

```
LD R0,($8000)
```

というプログラムは，実は同じ動作をするプログラムになるのです。もちろん，

```
LD (A0),R0
```

と書いて，レジスタ→メモリと転送することも可能です。

演算命令

転送命令の次は演算命令です。BUGの演算命令はおよそ次のようなものがあります。

加算（足し算）
減算（引き算）

図2　Z80と68000のレジスタ

Z80のレジスタ構成

A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'
IX			
IY			
SP			
PC			
I	R		

68000のレジスタ構成

a0	d0
a1	d1
a2	d2
a3	d3
a4	d4
a5	d5
a6	d6
	d7
USP	
SSP	
PC	
SR	

図3　BUGのレジスタ構成

それでは順番に説明していきましょう。

●加算命令

加算命令の代表が，ADD命令です。まず，下の例を見てください。

ADD R0, 1

これは，R0レジスタに1を加える命令です。BASICで書けば，R0=R0+1となるのでしょう。基本的なところは先ほどの転送命令と同じで，LDの代わりにADDを使えばいいのです。つまり，

ADD R0, R1

は，R0=R0+R1となるのだし，

ADD R0, ($8000)

と書けば，R0と$8000番地のデータを足し合わせる命令になります。こうなれば当然，アドレスレジスタを使って，

ADD R0, (A0)

と書くこともできるでしょう。

もちろん，アドレスレジスタにも加算命令は使えます。

ADD A0, 1

と書けば，A0=A0+1になります。

●減算命令

加算命令がADDなら減算命令はSUBです。使い方はADDと同じで，

SUB R0, 1

で，R0=R0−1になりますし，

SUB R0, R1

とすれば，R0=R0−R1になります。同様に，

SUB R0, ($8000)

で，R0から$8000番地に格納されている数値を引き算しますし，

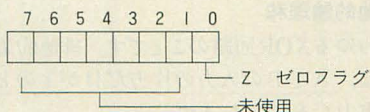
SUB R0, (A0)

で，R0からアドレスレジスタA0の示すメモリアドレスに格納されている数値を引き算します。もちろん，アドレスレジスタの減算も可能で，

SUB A0, 1

で，A0=A0−1となります。

例題2　次のプログラムを実行したとき，以下の問いに答えよ

図4　BUGのフラグレジスタの中身

リスト1　ジャンプ命令とラベル

```
      LD  R0, 1
LOOP: ADD  R0, 1
      LD  ($F000), R0
      JP  LOOP
```

```
LD   R0, 1
LD   R1, 5
LD   A0, $8000
ADD  R0, R1
SUB  R1, 2
ADD  A0, 1
```

問　レジスタR0，R1，A0に格納されている値を示せ

解答

R0=6, R1=3, A0=$8001

ジャンプ&サブルーチンコール

さて，マシン語にだってGOTO命令やGOSUB命令のような，ジャンプやサブルーチンコールなどの命令があります。それぞれ説明していきましょう。

●ジャンプ命令

BUGでは，JPがジャンプ命令です。プログラム中で，

JP $8000

と書けば，コンピュータは実行アドレスを$8000に移して，そこからプログラムを実行するわけです。まず，リスト1を見てください。リストの最後の行に，JP命令が使われています。しかし，JP命令の後ろには実行アドレスの代わりに，LOOPという文字が見えます。そう，ラベルです。X-BASIC以外のBASICではお馴染みですね。プログラムを見ると，2行目の先頭にちゃんと，LOOPという文字が見えます。このプログラムを実行すると，コンピュータがJP LOOPを実行したとき，実行はリストの2行目に戻るわけです。

●サブルーチンコール

サブルーチンコールはCALL命令を使います。使い方はジャンプ命令と同じで，

CALL $8000

と実行すれば，$8000番地以降に書かれたプログラムをサブルーチンプログラムとして実行していきます。JP命令と違うところは，RET命令を実行したときに，実行が元のアドレスに復帰することです。これはちょうど，BASICのGOSUB~RETURNと同じです。

例題3　次のプログラムを実行したとき，以下の問いに答えよ

```
      CALL SUB
      JP   JMP
SUB:  LD   R0, 3
      RET
JMP:  ADD  R0, 4
```

問　R0に格納されている数を答えよ

解答　7

条件分岐

IF~THEN~のような条件分岐は，別に高級言語の特権ではありません。マシン語だってちゃんとできるのです。それにはまず，フラグレジスタの仕組みを理解してもらいましょう。

BUGのフラグレジスタは8ビットのレジスタになっていますが，使っているのは第0ビットだけです。図4を見てください。レジスタの第0ビットがZゼロフラグとなっています。ゼロフラグは，計算結果が0になったときに1になります。たとえば，

LD　R0, 1

SUB R0, 1

と実行したとします。1行目でR0に1が代入されているので，SUB R0, 1と実行したら，R0の中身は0になります。そのときにフラグレジスタのゼロフラグが1になるのです。

また，フラグは命令を実行するたびに刻刻と変化していきます。そのため，

LD　R0, 1

SUB R0, 1

ADD R0, 1

と，実行したとすると，R0の内容は2行目で一度0になって，ゼロフラグが1になりますが，3行目で再びR0に1が加算され，R0の内容は0でなくなります。よって，3行目まで実行が終わった段階で，ゼロフラグは0となります。

Z80や68000などのCPUでは，このゼロフラグのほかにも，計算の繰り上がり，繰り下がりを示すキャリフラグや，オーバーフローの発生を知らせるオーバーフローフラグなど，たくさんのフラグが存在しますが，なにしろBUGは仮想CPUですから，もっとも単純なゼロフラグだけ用意しました。

さて，そのフラグの使い方ですが，条件付きのジャンプ命令というものが存在します。ゼロフラグに関するものは次の2つです。

JP　Z, $8000

これは演算結果が0だったなら（ゼロフラグが1なら）$8000へジャンプしなさいという命令です。そしてもうひとつが，

JP NZ, $8000

これは演算結果が0でないなら（ゼロフラグが0なら）$8000へジャンプしなさいという命令です。この2つの命令で，条件分岐を行うことができます。

例題4　次のプログラムを実行して以下の問いに答えよ

```
        LD   R0,0
        LD   R1,10
LOOP：ADD R0,2
        SUB  R1,1
        JP   NZ,LOOP
```

問　R0，R1の値を答えよ
解答　R0=20，R1=0

その他のマシン語の動作

さて，いままではマシン語の概念を説明するうえで，BUGという仮想マシンを用意して話を進めてきました。途中，要所ごとに例題を用意しながら話を進めてきましたが，これらの例題が解けた人は，もうマシン語の概念は理解できたと思っていいでしょう。

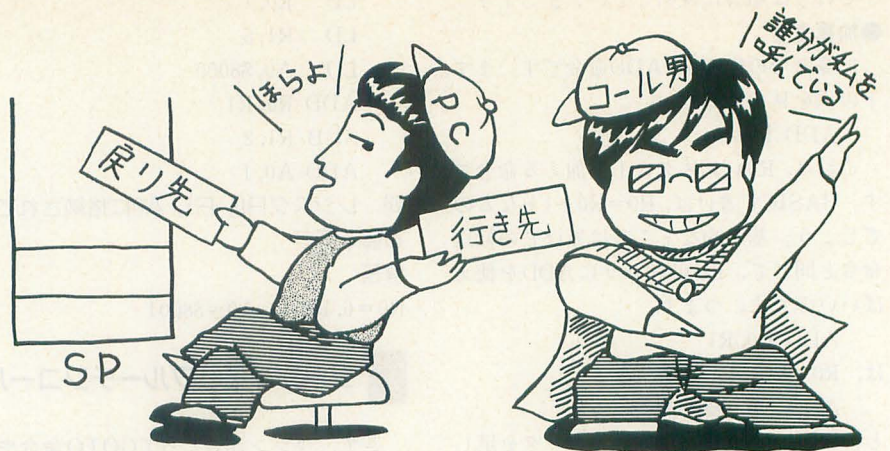
さて，ここらへんで一度BUGを使ったマシン語の解説は終わりにして，これからはもう少し違った角度からマシン語の説明をしていきましょう。

●スタックの動作

スタックという言葉を聞いたことがあるでしょうか？　スタックとはLIFO形式の構造を持ったデータ領域のことです。LIFOとはLast In First Outの略で，最後に入れたものを最初に取り出すデータ形式のことをいいます。

図5をご覧ください。まず，図のような箱を用意して，その中に球をひとつずつ落としていきます。図では最後に白い球を落とすことになっています。さて，これらの球は最初に箱に入ったものから順番に重なっているので，箱に全部球を入れたあとで箱から球を取り出そうとしたら，最後に入れた白い球を一番最初に取り出すことになります。つまり，最初に入れた二重丸の球を取り出すのは，一番最後になるのです。

このような構造をLIFO構造と呼びます。先ほども話したように，コンピュータのスタックは，このLIFO構造をとります。コンピュータのスタックは，SPレジスタ（スタックポインタレジスタ）の役割から理解しなくてはいけません。

SPレジスタはアドレスレジスタの一種で，その内容は常にメモリ上のあるアドレスを示しています。たとえばSPの内容が，$F000だったとします。さてそこへ，コンピュータからスタックへ$1234というデータが送られてきたとします。すると，アドレスを2引いた$EFFEへ，いま転送されてきたデータが格納されるわけです。ただし，送られてきたデータは2バイトですから，スタックのメモリも2バイト必要です。

すると$1234というデータの半分は，隣の$EFFF番地へ格納されるのです。そう，スタックデータはだんだん若いほうのアドレスへと格納されていくのです（普通は）。

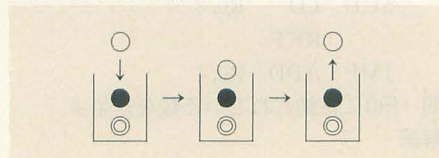
データをスタックにひとつ格納したとき，SPレジスタの値は$EFFEに変わっています。ここで，$8888というデータをスタックに格納すると，SPレジスタの値はさらに2つ減って$EFFCとなります。ここで，スタックからデータを取り出せば，最初に$8888，その次に$1234の順でデータが出てくるわけです。

さて，それではそのスタックはいったいどんなところで使われるのでしょうか？　まず，さりげなく使われているのがサブルーチンコールをしたときのリターンアドレスの格納です。Z80でCALL命令を実行したとき，CPUは現在実行しているアドレス（の次）を調べ，そのアドレスデータをスタックに格納しているのです。そしてRET命令を実行したときに，スタックから取り出したアドレスへ実行を移しているのです。

これは68000でも同じです。68000にはUSP，SSPと2つのスタックポインタが存在しますが，SSPはOSが，USPはユーザーが使うのが普通です。なお，68000ではCALLの代わりにbsr命令，RET命令の代わりにrts命令が使われていますが，名称が違うだけで動作にはなんの違いもありません。

さて，そのほかにもZ80には直接スタックを扱うPUSH，POPという命令があります。たとえば，

```
    PUSH HL
```

と実行すれば，HLレジスタペアの内容がスタックに格納され，

```
    POP HL
```

と実行すれば，スタックからHLレジスタペアにデータが転送されます。68000ではこのPUSH，POPに相当する命令はありませんが，データ転送命令（move命令）でスタックを操作することができます。

●論理演算の動作

さて，もうひとつコンピュータに欠かせないのがこの論理演算です。コンピュータで扱う論理演算には次の3種類があります。

論理積　　　　(AND)
論理和　　　　(OR)
排他的論理和　(XOR)

これらはそれぞれ、どのようなものなのでしょうか？　まずはそれぞれの基本動作から見ていきましょう。

基本的な論理回路は，図6のように2つの入力とひとつの出力回路を持っています。この2つの入力のパターンによって，出力が変わるのです。

●論理積

AND回路のことです。論理積はAとB2つの入力が両方とも1のときだけ，出力Cを1にします。

●論理和

OR回路のことです。論理和はAとBの入力のどちらか一方でも1なら，出力Cを1にします。

●排他的論理和

いわゆるXOR回路のことです。排他的論理和は，AとBの入力の片方だけが1のとき，出力Cを1にします。

コンピュータでは扱うデータをよく，2

図5　LIFOの動作

図6　論理回路の入力と出力

進数に置き換えて考えますが，この論理演算を行うときも2進数にして考えればわかりやすいでしょう。

たとえば，9は，2進数で表すと，1001B，12は1100Bとなります。ここで8と12の論理演算を行ってみましょう。まず，論理積からです。およそ下のような計算式になるはずです。

```
      1001B
AND)  1100B
-----------
      1000B
```

ここで見てもらえればわかると思いますが，コンピュータの論理演算は，2進数のデータの各桁同士で理論演算を行っています。2進数で1000Bというのは8のことですから，9 and 12＝8となります。同じようにORとXORについても計算してみましょう。

```
      1001B          1001B
OR)   1100B   XOR)   1100B
-----------   -----------
      1101B          0101B
```

このようにして見ると，9 or 12＝13，9 xor 12＝5となるのがわかるでしょう。マシン語を使うときの実際の命令では，レジスタを相手にするので，ビット数はもっと多いのですが，基本的な動作はいま示したとおりで，各桁ごとの論理演算を行っているのです。

簡単なプログラムの設計

さて，最後にいままで覚えてきたことのまとめとして，簡単なプログラムの設計をしてみましょう。いままで覚えてきた命令は，転送命令，演算命令，ジャンプ&コール命令，条件付きジャンプ&コール命令の4つでした。これらを使って，掛け算のプログラムを作ってみましょう。

さて，掛け算とはいったいどのような計算でしょうか？　掛け算の意味を調べればわかりますが，A×BはAという数をB回足し算するという意味の式です。足し算の命令は確かADDでした。そしてそれを繰り返すには条件判断付きのループを作ればいいのですから，簡単です。すると，掛け算部分のプログラムは次のような感じになるはずです。ちなみに，プログラムはBUGで書いてあります。

```
      LD   R0,0
LOOP: ADD  R0,R1
      SUB  R2,1
      JP   NZ,LOOP
       :
```

これはR0＝R1×R2をプログラムしたリストになります。さらに，KAZU1，KAZU2，KOTAEの3つのデータエリアを設けて，データをそこに格納するようにしてみます。つまり，

KOTAE＝KAZU1×KAZU2

というプログラムにするのです。そして，プログラムの終わりにRET命令を付ければ，立派なサブルーチンプログラムとして完成します。リスト2がその完成したプログラムです。

なお，リスト3に同じプログラムのZ80版，リスト4に68000版を載せておきましたので，比較してみるといいでしょう（誰ですか？　こんなプログラムを作らなくても，68000にはmul命令があるなんていっているのは？）。

*　　*　　*

いやー，こうしてほとんど駆け足でマシン語プログラムの概略を説明してきましたが，皆さんちゃんと理解してくれたでしょうか？　今回は私の独断と偏見により，BUGなる仮想マシンまででっちあげてしまって，「かえってわかりにくかった」などという人も出てくるのではないかと心配しています（すべて私の不徳のいたすところです）。

マシン語はとても楽しい言語です。ときには暴走したりして困ることもあるけど，なんといってもマシン語を触っていれば，コンピュータに一番近い言語を使っているという，ある種の優越感（？）に浸れます。皆さんもこれを機会に，マシン語人生をエンジョイしてみませんか？

ソースジェネレータを使う

家電製品を買ってくると，まず分解して中身を見るという人がいます。初心者には決しておすすめできませんが，ハードウェアの知識があれば，それだけで結構楽しめるようです。

ソフトウェアについても同様のことがあります。サブルーチン程度ならともかく，1本の完結したマシン語プログラムを書くのはけっこう大変な作業です。マシン語の勉強をする際には，ちゃんと動くマシン語プログラムに少しずつ手を入れていくという方法をとることが有効となる場合があります。

グラフィックデータを引っ張ってきたり，音源ドライバ用のデータをデコードしたり，なにか変わったことをやっていたら，とりあえずなにをしているのかと覗いてみる，ゲームを無敵モードにしてしまうなどというのもマシン語入門の過程で一度は通る道でしょう（通るだけで長居をするもんじゃあないが）。

こういうときに便利なのがソースジェネレータです。S-OSならRING，X68000ならDIS.Xというところでしょうか。これらは実行形式のプログラムからアセンブラにかけることのできるソースリストを作成してくれます。プログラムには見分けにくいデータエリアなども手動/自動で設定できるなど，どちらもなかなか高性能なツールです。

このようなツールを使って作られたソースプログラムにはマシン語プログラミングのノウハウがいっぱいに詰まっています。まずは小さなツールからソースジェネレートして処理を追ってみるのがよいでしょう。よいソースプログラムを読むこと以上の勉強法はないのですから。

多くの場合，ソースジェネレータを使ってもラベル名やコメントなどは復活されません。Oh!X誌上に掲載されているものならダンプリストを打ち込み，ソースジェネレート後，ラベルやコメントを整えていくというのがよいでしょう。フリーウェアなどプログラムの著作権的に問題のなさそうなもので，かつ動作のわかったサブルーチンなどは切り出してライブラリ化などすれば，貴重な知的財産になります。そして次のステップに進んでください。

リスト2　掛け算プログラム（BUG用）

```
PMUL:  LD   R0,0
       LD   R1,(KAZU1)
       LD   R2,(KAZU2)
LOOP:  ADD  R0,R1
       SUB  R2,1
       JP   NZ,LOOP
       LD   (KOTAE),R0
       RET
KAZU1:DEFS  1
KAZU2:DEFS  1
KOTAE:DEFS  1
```

リスト3　掛け算プログラム（Z80用）

```
PMUL:  LD A,0
       LD C,(KAZU1)
       LD B,(KAZU2)
LOOP:  ADD A,C
       DJNZ LOOP
       LD (KOTAE),A
       RET
KAZU1:DEFS  1
KAZU2:DEFS  1
KOTAE:DEFS  1
```

リスト4　掛け算プログラム（68000用）

```
PMUL:  move.l  #0,d0
       move.l  KAZU1,d1
       move.l  KAZU2,d2
LOOP:  add.l   d1,d0
       sub.l   #1,d2
       bne     LOOP
       move.l  d0,KOTAE
       rts
KAZU1:ds.l   1
KAZU2:ds.l   1
KOTAE:ds.l   1
```

MC68000の動作を探る

宮島 靖 Miyajima Yasushi
鈴木 康弘 Suzuki Yasuhiro

マシン語とはCPUに対する直接的な命令です。CPUの動作を理解することがマシン語の理解であるともいえます。ここでは68000の基本的な命令とそれがどのように動いているのかを解説します。マシン語の動きをCPUの動作から見てみましょう。

CPUの世界

コンピュータの内部はとてつもなくデジタルな世界で、電圧が高いか低いかだけで動作しています。電圧が高いものを1，低いものを0とすると，0と1の2進数の世界になります。

CPUもデジタルなやつなので，0と1の組み合わせで動くように作られています。したがって，CPUになにか仕事をしてもらいたいときには，0と1の組み合わせで指示するしかありません。この0と1の組み合わせは，俗にいうマシン語のプログラムというものです。2進数8桁を16進数2桁で表すと，よく雑誌で見かけるダンプリストになります。

巷では，2進数8桁を8ビット，2進数16桁を16ビットなどと呼んだりします。16進数で2桁の数字は，8ビットです。8ビットをまとめて1バイトと呼んだりもします。

CPUは，このマシン語でしか動作しないので，BASICやCで書かれたプログラムもマシン語に変換されて（変換するのもマシン語のプログラム）実行されているのです。

メモリ

メモリとは，俗にいうRAM (Random Access Memory) やROM (Read Only Memory) のことです。メモリの中には，マシン語のプログラムや，いろいろなデータなどが入っています。メモリの最小単位は通常1バイト（=8ビット）です。そして，先頭から順番に番号がつけられています。一番先頭を0番地，その次を1番地，というように，番地という言葉を使って，場所を指定します。これはBASIC風に説明すると，メモリとは1つの要素が1バイトの大きさの配列変数みたいなものです。配列変数の添字が番地にあたります。68000ではこの番地に000000H～FFFFFFHまでを指定することができます。

しかし，CPUが基本的にできる作業は，整数の代入，四則演算，論理演算，条件判断，分岐くらいしかありません。

・ですから，プログラムをマシン語で組もうと思うと，画面に文字を出すだけでも大変な作業になります。BASICのようにprint命令だけでなんでも表示できるようなお手軽さは，マシン語にはないわけです。

ですから，他人の作ったプログラムもちょっと見ただけではなにをしているのかわからず，1つひとつ追ってみたら文字を表示するプログラムだったなどということもあります。

アセンブリ言語

では，マシン語のプログラムを作るということは，CPUに与える2進数の組み合わせを作るということなのでしょうか。確かに，その2進数の組み合わせを作ってもプログラムはできてしまいますが，2進数の羅列を直接入力していたのでは，画面に文字を表示するだけのプログラムでさえ挫折してしまうことでしょう。

では，実際にどのように開発を行うかというと，アセンブリ言語と呼ばれる言語でプログラムを書いていきます。アセンブリ言語がBASICやC言語などと根本的に違うのは，CPUが理解できる命令（先ほど説明したように，2進数の羅列ですよ）に1対1で対応しているというところです。

レジスタ

CPUは，レジスタという，BASICでいう変数みたいなものを持っています。BASICの変数と異なる点は，BASICの変数はプログラマが自由に名前をつけられ，メモリの許す限りいくつでも作れ，浮動小数点も自由に扱うことができるのですが，レジスタはCPUによって数と名前が決まっており，整数しか扱うことができないということです。

68000の持っているレジスタは，データレジスタ8本，アドレスレジスタ7本，プログラムカウンタ1本，スタックポインタ2本，ステータスレジスタ1本の19本です。ステータスレジスタが16ビットなのを除くと，ほかはすべて32ビットレジスタです。

データレジスタは，主に演算などに使用されるもので，BASICで使用する変数と，役割はほとんど同じです。D0，D1，……D6，D7の8本あります。

アドレスレジスタは，メモリをアクセスする際に，番地指定のために使われるレジスタです。A0，A1，……A5，A6の7本です。

プログラムカウンタは，次に実行される命令の入っている番地を指します。マシン語のプログラムはメモリに入っているわけですが，メモリにはほかのプログラムや，データなどもたくさん入っています。

CPUは，このプログラムカウンタが指す番地の内容を命令とみなし，プログラムを実行します。

また，命令を読み込むと，プログラムカウンタは次の命令の番地を指すようになっているので，CPUは常にプログラムカウンタが指す番地から命令を取ってくるだけでよいのです。プログラムカウンタはPCと表記します。

スタックポインタとステータスレジスタはあとで説明します。

プログラムの終了

BASICでは，end命令を実行した時点でプログラムが終了します。また，endがなくてもプログラムの最後（次に実行するも

のがない）まで行くと，自動的にプログラムが終了します。

しかし，マシン語ではきちんとプログラム終了の命令を書いておかないといけません。なぜなら，CPUはプログラムの最後がわからないからです。マシン語のプログラムはメモリ上に置かれ，先頭からPC（プログラムカウンタ）が順番に命令を指して実行されていきます。

プログラムの最後にきても，PCはその次の命令を指すようになっているので（プログラムの最後はプログラマがプログラムした最後ということであり，CPUはおかまいなしに爆走している），CPU はその次の命令を実行しようとしてしまいます。そこにどんな命令があるかは，プログラムを作った人にはわからないので，どんな動作をするかが見当がつきません。俗にいう暴走という状態になってしまうのです。

で，プログラム終了の命令なんですが，次の行をプログラムを終了させたいところに書いてください。

```
dc.w   $FF00
```

これがなにを意味するかはあとで説明しますから，いまのところは気にせず,end命令なんだなということにしといてください。

なにか作ってみるぞ

では，とりあえずレジスタになにか数字を入れるプログラムを作ってみましょう。

```
move.l   #0,d0
move.w   #1,d1
move.l   #2,a0
dc.w     $FF00
```

これは，D0レジスタに0，D1レジスタに1，A0 レジスタに2を代入するプログラムです。

“move”というのは，代入の命令です。“.l”というのは，扱うデータの幅を32ビットにする，という意味です。“#0”は数字の0を意味します。アセンブラでは，数字を表すときはこのように数字の前に“#”をつけます。“d0”は，レジスタの D0 レジスタのことです。

“#0,d0”の部分をオペランドといい，68000では左の“#0”をソースオペランド，右の“d0”をデスティネーションオペランドといいます。

“move” 命令は，ソースオペランドをデスティネーションオペランドに転送するという命令なので，この場合0がD0 レジスタに入ります。つまり，D0レジスタの内容を0にするのです。

BASIC風に書くと，

d0＝0

みたいな感じです。

“.l”の部分ですが，これはソースオペランドとデスティネーションオペランドのサイズを指定しているのです。“.l” は32ビットのことです。68000は，数値を3つのサイズで表せるようになっています。ロングワード（32ビット），ワード（16ビット），バイト（8ビット）の3種類です。この場合，ロングワードで転送が行われ，D0レジスタの全32ビットが影響を受けます。もし，ワードの指定ならD0レジスタの下位16ビットのみ影響を受け，上位16ビットは影響を受けません。

ワードを指定するときは，“.l”の代わりに“.w”を書きます。また，バイトを指定するときは，“.b”を書きます。

次の文は，1をD1レジスタに入れる命令です。ただし16ビットという指定があるので，結局D1レジスタの下位16ビットに1が入ります。D1レジスタの残りの上位16ビットは影響を受けず，この命令を実行する前の状態のままになります。

その次の文は，2をA0レジスタに入れる命令です。ちなみに，アドレスレジスタは必ずロングワード（32ビット）で実行されます。アドレスレジスタは，番地を指定するために存在するレジスタですので，ワードやバイトの操作は無意味というわけです。では，ワードで値を代入するとどうなるかというと，32ビットに符号拡張されてからロングワードで処理が行われます。

符号拡張というのは今回は説明を省きますが，要するに32ビットの値にするということです。この符号拡張の意味を理解しないで使わないほうがよいでしょう。32ビットに符号拡張されてから実行が行われるので，ワードでもロングワードでもCPU内部の処理速度は同じです。あと，バイトの操作はできません。ロングワードもしくはワードのみです。

さて，このプログラムを実行させてみましょう。このプログラムのファイル名をTEST.Sにしたのなら，

```
AS TEST.S
LK TEST.0
```

としてください。これで，ディスクに TEST.X という実行ファイルが作られます。実行は，

```
TEST
```

とするだけでOKです。

実行すると，すぐプロンプトが出てきますね。プログラムは実行されたのでしょうか？

図1

このプログラムの場合，レジスタに値を入れるだけで，ほかにはなにもしません。したがって，レジスタに値を入れたあと，すぐプログラムを終了してHumanに戻ってきてしまいます。

では，レジスタの値を表示させましょう。でも，マシン語には，BASICみたいなprint命令はないのです。そこで，DOSコールの登場です。さ，プログラマーズマニュアルを引っ張り出して，手元に置いておきましょう。

DOSコール

マシン語には，文字を表示する命令はないと書きましたが，文字を表示できないわけではありません。実際にいろんなところで文字を表示していますし。

実は，X68000には，テキストVRAMというRAMがあり，ここに表示したい文字のデータを書くことによって，ハードウェアでその文字を画面に表示する仕組みになっているのです。したがって，なにか文字を表示させる場合は，このテキストVRAMにその文字のデータを書き込んでしまえばよいのです。この作業はプログラムにすると，結構面倒臭いものです。

テキストVRAMの構成は図2のようになっています。これは普通の16ビット機のグラフィックに相当する領域ですから，ここに適当なデータを書き込むと画面になんらかの模様を描くことができます。

まず，表示する位置を指定し，それを対応するVRAMのアドレスに変換します。次に表示したい最初の文字を取り出し，漢字コードから漢字ROM内のフォントデータ格納アドレスを計算して，最初のデータを先ほど求めたアドレスに書き込みます。さらに1ライン下のアドレスを求め，そこに次のデータを書き込んでいきます。これを16ライン分繰り返してやっと1文字表示の終わりです。

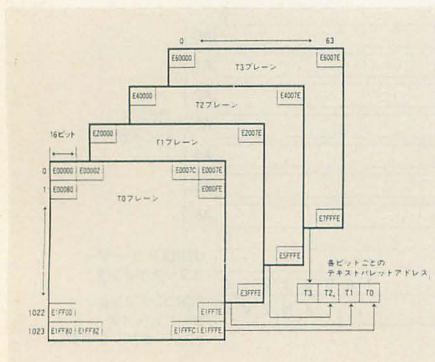



図2

もし，文字に色がついていたら対応するプレーン分の処理を行い，強調指定があればデータをシフトさせたものと論理和をとって書き込んでいきます。さらに外字が使ってあったときのことも考えなければなりませんから，実際にはもっと面倒な処理が必要です。真っ先に外字にあたるコードかどうかを調べ，外字ならシステムの外字エリアを順に検索してフォントデータを得ることになります。

なにか文字を表示させるプログラムを作るたびに，この文字表示プログラムを作っていたのでは，夜が明けてしまいますね(プログラムは夜中に作ることが多い)。でも，実はあるんです。この文字表示プログラムが。

どこにあるのかというと，それはHumanの中です。ある方法で，このHumanの文字表示プログラムを簡単に呼び出すことができ，それを使うと，いとも簡単に文字を表示することができます。文字表示だけでなく，ディスクアクセス，キー入力など，いろいろなものが用意されています。これらを，DOSコールと呼びます。このDOSコールを自在に使えるようにならなければ，アセンブラでの開発は不可能です。ちなみに，先ほど説明した，プログラムの終了の命令の，

```
dc.w   $FF00
```

ですが，これもDOSコールです。

スタック

さて，DOSコールを使えるようになるために必要な知識として，スタックがあります。このスタックは，普通にプログラムを組むときにも重要な概念なので，必ず理解しておいてください。

レジスタの説明のところで，スタックポインタなるレジスタが出てきましたが，このスタックポインタが今回の主役です。2つあると書きましたが，普通はそのうちのひとつしか使えません。

もうひとつは，HumanなどのOSが使っているのです。この，自分が自由に使えるスタックポインタを，SPまたはA7と書きます。アドレスレジスタはA0～A6で，スタックポインタはA7になりますが，このA7レジスタはほかのA0～A6とは性格が異なりますので注意してください。以降，スタックポインタは，SPと表記します。

さて，スタックとはいったいなにものでしょうか。スタックを説明するために，とりあえず以下のBASICのプログラムで説明します。

```
10 int mem(9999)
20 int sp=10000, x, y
30 push(10)
40 push(20)
50 x=pop( )
60 y=pop( )
70 print x-y
80 end
90 func push( x:int )
100   sp=sp-1
110   mem(sp)=x
120 endfunc
130 func pop( )
140   int x
150   x=mem(sp)
160   sp=sp+1
170   return(x)
180 endfunc
```

このプログラムを実行させると，10を表示します。なぜ10を表示するかがわかると，もうスタックを半分理解したのと同じです。

関数pushは，spの値を1減らして，mem(sp)に引数を格納する関数です。

30行目のpush(10)の場合，spは最初10000ですから，sp=9999になり，mem(9999)=10となります。

40行目のpush(20)の場合も同様に，sp=9998になって，mem(9998)=20となります。

関数popは，返り値にmem(sp)をセットして，spを1加える関数です。

50行目のx=pop( )の場合，spの値は9998ですので，返り値がmem(9998)になります。これは40行目で代入した20です。その後，spに1が加えられ，sp=9999になります。したがって，xには20が入ります。

60行目のy=pop( )の場合も同様に返り値は30行目で代入した10になり，sp=10000となります。したがって，yには10が入ります。ここまででx=20，y=10が入って

いるので，70行目のx－yで10が表示されるのです。

関数pushを次々と呼ぶと，呼んだ順番のとおりにそれぞれの引数が，配列変数memの最後の要素から入っていきます。関数pushの処理が終わったときのspの値は，最後にpushした添字になっています。

また，関数popを呼ぶと，pushによって入った値を次々と取り出していきます。関数popの処理が終わったときのspの値は，次にpopで読むところの添字になっています。

関数pushと関数popの処理が終わったときのspの値は，要するに有効なデータが入っている先頭を示しているということです。

なにを面倒臭いことを，と思っているでしょうが，これこそスタックなのです。配列変数memが，実際のメモリ（添字は番地になります），変数spがスタックポインタになります。

要するに，スタックというのは，メモリのある番地に，データを代入していったり，そこからデータを取り出していったりすることなのです。次々に値を代入していっても，スタックポインタがどんどん移動していきますので，pushによって以前にpushで代入した値が壊れることはありません。

この，pushの動作を，スタックに積む，といいます。先ほどのBASICのプログラムの30行目では，10をスタックに積んでいるのです。

このpushとpopの動作をするのは，アセンブラでは1命令になります。たとえば，10をスタックに積みたいときは，

```
move.l  #10, -(sp)
```

と書きます。メモリはひとつが8ビットです。この場合，32ビットで行われるので，番地が4つ分必要です。したがって，SPは4減ります。また，レジスタもスタックに積むことができます。

```
move.l  d0, -(sp)
```

で，D0レジスタの内容がスタックに積まれます。

スタックから値を取り出したいときは，

```
move.l  (sp)+, d0
```

と書きます。この場合，スタックに入っていた値がD0レジスタに入ります。

さて問題です。次のプログラムの実行終了後，D0レジスタにはなにが入っているでしょうか？

```
move.l  #1,d0
move.l  d0,-(sp)
move.l  #2,d0
move.l  (sp)+,d0
```

答えは簡単ですね。1です。1行目で1をD0レジスタに入れ，2行目でD0レジスタをスタックに積んでいます。だから，スタックには1が積まれます。3行目で2をD0レジスタに入れているのですが，4行目でスタックから取り出した値をD0レジスタに入れています。したがって，1が入るわけです。

この場合3行目の文は無意味なのですが，ここで2行目と4行目の文によってD0レジスタの値が保存されることに注意してください。つまり，2行目より前と，4行目以降では，D0レジスタの内容は等しいのです。これは，よくレジスタの値が壊れてほしくないときに使うもので，3行目がD0レジスタを使う意味のある文で，その前後でD0レジスタの値が変わってしまっては困るときに使います。

ちなみに，D0レジスタだけでなく，D1もD2レジスタも壊れてしまっては困る，というときは，D0, D1, D2レジスタをスタックに積んで，あとでD0, D1, D2レジスタをスタックから取り出せばよいのですが，スタックは一番最後に積まれたものが一番最初に取り出されるということに注意してください。つまり，

```
move.l  d0,-(sp)
move.l  d1,-(sp)
move.l  d2,-(sp)
```

でスタックに積んだら，取り出すときは，

```
move.l  (sp)+,d2
move.l  (sp)+,d1
move.l  (sp)+,d0
```

となります。

さらにおいしいことに，実は複数のレジスタをスタックに積んだり，取り出したりするのに便利な命令があるのです。

たとえば，D0, D1, D2レジスタをスタックに積みたい，と思ったら，

```
movem.l  d0/d1/d2,-(sp)
```

というように記述できるのです。また，こうして積まれたレジスタを復活させたいときは，

```
movem.l  (sp)+,d0/d1/d2
```

と書けるのです。

D0, D1, D2というように，レジスタの番号が続いたものを記述するときは，

```
movem.l  d0-d2,-(sp)
movem.l  (sp)+,d0-d2
```

でもOKです。

続・DOSコール

では，早速DOSコールを用いてなにか作ってみましょう。

DOSコールでスタックが必要なのは，DOSコールで引数を渡すのにスタックを用いているからなのです。たとえば，1文字表示のDOSコールに，表示させたい文字を渡すのには，表示させたい文字をスタックに積んでDOSコールを呼べばいいのです。

1文字表示のDOSコールは，プログラマーズマニュアルを見ると，$FF20の_PUTCHARです。コールのところを見ると，

```
move.w  CODE, -(sp)
dc.w    _PUTCHAR
addq.l  #2,sp
```

と書いてあります。1行目は，表示したい文字コードをワード（16ビット）でスタックに積むことを表しています。2行目で，1文字表示のDOSコールを呼んでいます。これは，DOSCALL.MACを使わないならば，

```
dc.w  $FF02
```

と書いても同じことです。

さて，3行目はなんでしょう。これは，SPに2を足すという命令です。アセンブラマニュアルに，"addq"命令の説明が載っていますので，見ておいてください。要するに，デスティネーションオペランド（sp）に，ソースオペランド（#2）を，ロングワードで足すということです。

これはなにをしているのかというと，1行目でワードで文字コードをpushしました。そのため，文字コードをpushする前に比べてスタックの値が2減っているのです。したがって，2を足しておかないと，スタックが文字コードのところを指したままになってしまうのです。

たとえば，次のプログラムを見てください。

```
move.w  d0,-(sp)
move.w  #$41,-(sp)
dc.w    _PUTCHAR
move.w  (sp)+,d0
```

これは，DOSコールの前後でD0レジスタをワードで保存しているつもりですが，DOSコールのあとのスタックの補正をしていないため，4行目では$41がD0レジスタに入ってしまうことになります。

ちなみに，$41ですが，これは16進数の41を表しています。16進数を表すときは，数字の前に$をつけるのです。これはASCIIコードで'A'を表しています。

では，文字を表示するプログラムを作ってみましょう。とりあえず，'A' を表示させたいのならば，まず'A' の文字コードである$41をスタックに積みます。そして，DOSコールをして，次にスタックに2を足して，スタックポインタの値を元に戻しておきます。

```
move.w  #$41,-(sp)
dc.w    _PUTCHAR
addq.l  #2,sp
dc.w    _EXIT
```

プログラムの最後に，プログラム終了のDOSコールを入れるのを忘れないでください。

このプログラムを実行すると，'A' が表示されます。この要領で，たとえば'X68000'を表示するプログラムを作ってみてください。

フラグ

さて，たとえば，D0レジスタが1だったら'1'を，1でなかったら'0'を表示させるプログラムを作りたくなったとしましょう。1を表示したり0を表示したりするのは，先ほどのDOSコールで実現できますが，問題はD0が1だったら，というところにあります。なんとかしてD0が1かどうかを調べなければなりません。

そこで，レジスタのところで最後に出てきた，ステータスレジスタを利用することになります。このレジスタは別名フラグレジスタといい，フラグという1ビットのレジスタが集まってできたものです。フラグの中で，特に重要なのがゼロフラグと，キャリフラグと呼ばれるものです。

ゼロフラグというのは，演算の結果が0になったときにセットされ(1になる)，0以外のときはリセットされます(0になる)。たとえば，

```
sub.l  d0,d0
```

という命令で（ちなみに"sub"というのは引き算をする命令で，デスティネーションオペランドーソースオペランドがデスティネーションオペランドに入ります)，この結果D0レジスタは0になり，ゼロフラグがセットされます。また，

```
sub.l  #1,d0
```

という命令で（これはD0レジスタの内容から1を引く命令です)，1を引いた結果が0ならばゼロフラグはセットされ，0でなければリセットされます。

次にキャリフラグですが，これは演算の結果で桁上がり，または桁下がりが起こるとセットされ，それらが起こらなければリセットされます。わかりやすくいうと，キャリフラグは演算の結果が表せる数の範囲を越えるとセットされます。たとえば，バイトの演算では，表せる数は0〜255です。したがって，加算をして255 を越えたり，減算をして0を下回るとセットされるのです。ところで，もし255を上回ったり，0を下回るような計算をすると，レジスタにはなにが入るのでしょう。たとえば，次のプログラムを見てください。

```
move.b  #$80,d0
move.b  #$80,d1
add.b   #$90,d0
sub.b   #$90,d1
```

これは，3行目で$80+$90，4行目で $80−$90を実行しています。両方ともバイトの指定がありますので，バイトで表せる範囲を越えています。したがって，両方ともキャリフラグはセットされます。これを実行したとき，D0,D1レジスタにはなにが入っているのでしょう。

まず，D0レジスタ ($80+$90) を考えてみましょう。普通に演算すると答えは$110です。こういうときは，下位8ビットの内容がD0レジスタに入ります。つまり，$10が入るのです。キャリフラグを第9ビット目と考えることができます。

では，D1レジスタ ($80−$90) はどうでしょうか。普通に演算すると，−$10です。実は，−$10というのは，$F0と表せられるのです。なぜかというと，8ビットの演算を考えた場合，$F0に$10を加えると，$00になるからです（キャリフラグがセットされますが)。$10加えて$00になるのですから，−$10と同じです。そういうわけで，D1レジスタには$F0が入ります。

ちなみに，同様に−$01は$FF，−$02は$FEとなります。負の数をこのように表記することを，2の補数表示といいます。

次に，条件ジャンプ命令です。これは，ある条件が成り立つときのみジャンプするという命令です。条件というのは，たとえばゼロフラグが1だったら，とかゼロフラグが0だったら，というものになります。ゼロフラグが1ならジャンプする，という命令は，

```
beq  番地
```

ゼロフラグが0ならジャンプする，は，

```
bne  番地
```

キャリフラグが1ならジャンプする，は，

```
bcs  番地
```

キャリフラグが0ならジャンプする，は，

```
bcc  番地
```

という命令になります。

これを使うと，最初の，D0レジスタが1ならば1を表示させ，それ以外なら0を表示させるプログラムは，次のようになります。

```
        sub.l   #1,d0
        beq     one
        move.w  #$30,-(sp)
        bra     other
one:
        move.w  #$31,-(sp)
otehr:
        dc.w    _PUTCHAR
        addq.l  #2,sp
        dc.w    $FF00
```

1行目で，D0レジスタから1を引きます。もしD0に1が入っていたら，D0レジスタは0になり，ゼロフラグがセットされます。また，D0に1以外が入っていたら，D0レジスタは0にはならず，ゼロフラグがリセッ

図3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
T		S			I2	I1	I0				X	N	Z	V	C	SR ステータスレジスタ

トされます。したがって，2行目で，D0に1が入っていたときは“one”に飛び，その他の場合はそのまま次の文が実行されます。

このプログラムの1行目で，D0レジスタが1かどうかを判断しているわけですが，このときD0の内容が書き換わってしまうことに注意してください。つまり，元々入っていた値よりも1少なくなっているのです。

引き算は，このようにある値を引くことによって，フラグを変化させることができます。しかし，レジスタに残る値は無意味です。この場合，D0レジスタの内容が1かどうかを調べたいだけで，D0レジスタに1少ない値が残る必要はないのです。どちらかといえば，D0の値はそのままのほうがいいでしょう。そこで，引き算をしてフラグは変化するけれど，引き算をした結果はレジスタに入らないという便利な命令があります。それは，“cmp”という命令です。たとえば，

```
	cmp.l	#10,d0
	beq	yahho
	cmp.l	#20,d0
	beq	haheho
```

のように使います。

このプログラムは，D0レジスタの値が10なら“yahho”に飛び，20なら“haheho”に飛ぶプログラムです。1行目でD0の値が変わらないので，3行目で20と比較するときもそのまま“#20”と書くことができます。これを“sub”命令を使って書くと，

```
	sub.l	#10,d0
	beq	yahho
	sub.l	#10,d0
	beq	haheho
```

となり，3行目で20と比較していることを理解するには1行目で10を引いていることを知っていなくてはならないのです。

アドレッシングモード

いままで，“#0”と書くと0を，レジスタ名をそのまま書くとそのレジスタを表す，などと書いてきましたが，実はほかにもいろいろあります。この，“#0”で0を表す，というのをアドレッシングと呼びます。ここでは，基本的なものを紹介します。

まず，数値をそのまま書くとその番地の内容を表します。

```
	move.b	$10000,d0
```

この例では，$10000番地の内容がD0レジスタに入ります。

次に，アドレスレジスタに括弧をつけると，そのアドレスレジスタが指す番地の内容を表します。

```
	move.l	#$20000,a0
	move.w	(a0),d0
```

これは，$20000番地の内容をワードでD0レジスタに入れます。ワードですので，$20000番地が上位8ビット，$20001番地が下位8ビットになります。

これの発展したもので，括弧の前に‘－’をつけるとアドレスレジスタがサイズ分引かれてからその内容を表し，括弧の後ろに‘＋’をつけるとその内容を表すのはその番地ですがそのあとサイズ分足されます。

```
	move.l	#$10000,a0
	move.l	#$20000,a1
	move.w	#100,-(a0)
	move.w	#200,(a1)+
```

この場合，3行目では$10000からまずワード分の2が引かれ，A0レジスタに$FFFEが入ったところで$FFFE番地に100が入ります。4行目では，まず$20000番地に200が入り，その後A1レジスタに2が足されます。

とりあえず，これだけ知っていれば大丈夫でしょう。ほかのアドレッシングモードはアセンブラに慣れてきてから自分で覚えていってください。

アセンブラプログラマへの道

ページ数の都合もあり，かなり急いで説明してきましたが，理解できたでしょうか。一度に全部理解しようとはせず，わかるところを少しずつ増やすような気持ちで勉強していきましょう。とにかくアセンブラをマスターするには，暇があったら，アセンブリ言語のプログラムを書いて，実行してみることです。そうすると，アセンブラ的な考え方がわかってきます。慣れてしまえばこっちのもんです。

アセンブラでプログラムが書けるようになると，プログラムはなんでもアセンブラで書きたくなります。こうなれば，いやでもアセンブラの技術は上がっていきます。とにかく，毎日アセンブラに触っていることです。

ジャンプ

さて，X-BASICでは禁じ手となっている，goto命令を知っているでしょうか。これは，

```
	goto	行番号
```

というように使い，この命令があると，その行番号にジャンプするのです。

たとえば，

```
	10 int a
	20 a=10
	30 goto 50
	40 a=20
	50 print a
	60 end
```

というプログラムの場合，30行にくると，50行に飛んでしまい，40行を実行しません。したがって，50行で表示されるのは10です。

アセンブラにも，こういう命令があり，実行番地を指定して，その番地に制御を移すことができます。この命令は“jmp”です。たとえば，

```
	jmp	$10000
```

と書くと，この行を実行したら，次は$10000番地から命令を実行していきます。

また，同じくある番地に飛ぶという“bra”という命令があります。これは，飛び先の番地が近いときにだけ飛ぶことができます。“jmp”命令より多少実行速度が速いので，とりあえず“bra”命令のほうを使います。

さて，普通アセンブラでプログラムを組むとき，どこの番地から実行されるなんてわかりません。したがって，“bra”命令でジャンプしたくても，ジャンプ先の番地はわからないのです。

そこで，アセンブラでは次のように書きます。

```
	move.w	#$41,d0
	bra	yahho
	move.w	#$42,d0
yahho:
	move.w	d0,-(sp)
	dc.w	_PUTCHAR
	addq.l	#2,sp
	dc.w	_EXIT
```

2行目では，番地を書く代わりに“yahho”と書いてあります。そして，4行目に“yahho:”というのがあります。これは，想像がつくと思いますが，2行目を実行すると，4行目にジャンプするのです。この“yahho”を，ラベルといいます。ラベルは，プログラム中の任意の行の先頭で，

```
	ラベル:
```

と書くことによって，そのラベルを使っているところでアドレスに置き換わります（置き換えているのは，AS.Xです）。したがって，もし5行目の文が$10000番地になったとしたら，2行目は

```
	bra	$10000
```

という文と同じになるのです。また，この場合で，

```
	move.l	#yahho,a0
```

というのがあれば，これは，

```
	move.l	#$10000,a0
```

と同じ意味になり，結果的に“yahho”の番地をA0レジスタに入れていることになります。

基本装備とおまじない

ここでは68000でマシン語を始める際の基本的なツールの使い方，そしてマニュアルの見方……などを中心にマシン語入門者の道案内をしていきます。最初はおまじないのような命令もやがて理解されていくことでしょう。

Komura Satoshi
古村 聡

ようこそ，マシン語の世界へ。だが，この世界での冒険へと旅立つ勇者よ，マシン語の世界もロールプレイングゲームの世界と同じようにそれなりの心構えとそれなりの装備が必要なのです。それにいろいろな手ごわい敵も潜んでいます。私はまだまだベテランとはいえないけれどそこそこ経験を積んでいるから役には立つと思いますよ。冒険へと向かうのなら私を仲間に入れませんか？

(で)を仲間にしますか？(Y/N) Y

(で)は仲間になりました。

冒険の前にまず，装備を整えましょう。では，まずマシン語のプログラムを作るのに必要なものを挙げていきましょうか。内容についての解説がわからなくても名前だけは覚えてそれぞれ揃えるようにしてください。

では，まずはソフトから。

●ED.X

Human68k用のスクリーンエディタ。標準のシステムディスクに入っています。またC CompilerPRO-68K(以下XC)，福袋V2のパッケージにも含まれています。

●AS.X

68000用のアセンブラ。初代機であれば福袋のディレクトリに入っています。その他のACEやPROなどでは別売りで「福袋V2」「XC」に入っています。

●LK.X

同じくリンカ。アセンブラと同じく初代の「福袋」，別売りの「福袋V2」「XC」にあります。上級者はリンカとしてLK.XではなくPDSのHLK.X(ハイスピードリンカ)などを使っている人も多いようですが，標準のLK.Xがあればとりあえず十分。

●DB.X

同じくシンボリックデバッガ。「福袋V2」「XC」にあります。

●DOSCALL.MAC

DOSコールのマクロが入っているファイル。「XC」にありますが，「福袋V2」はついていません。またOh!Xの先月号のディスクのなかにも入っています。

●IOCSCALL.MAC

同じくIOCSコールのマクロ。「XC」，それとOh!Xの先月号のディスクのなかにも入っています。

マニュアル類としては，

●プログラマーズマニュアル

PROCESS.X，WHERE.Xなどのユーティリティ類の解説，DOSコール，IOCSコール，日本語FPのファンクションコールの呼び出し方，BASIC外部関数，デバイスドライバ，ユーザープログラムの起動後の状態などについての解説書です。X68000でプログラムを作る際にはこれがないと始まりません。「XC」「福袋V2」に付属します。

●アセンブラマニュアル

プログラマーズマニュアルと同じく「XC」「福袋V2」に付属するマニュアルでアセンブラ，リンカ，デバッガなどの取扱説明書です。またアセンブラの命令セットリファレンスも掲載されています。

基本的にはこれらのものが必要です。よく見るとわかりますが必要なものはほとんど福袋V2かXCを買うとついてきます。つまりマシン語をやりたい人はとりあえず，

XCか福袋V2を買ってくる

ことが必要になるわけです。ま，どちらでも好きなほうを買えばいいでしょう。

マニュアルではないのですが，マシン語プログラムを組む際の参考資料として技術評論社の「68000プログラマーズハンドブック」や工学社の「X68000環境ハンドブック」などを買った人も多いようですよ。興味があるのなら，どういうものか見てみて気にいれば買ってみるといいでしょう。

冒険の書(その1)

お手ごろなところで「CRTのモードを512×512ドットにする」あたりからいってみましょうか(リスト1)。なに，初めてでも恐れることはありません。

まず，CRTモードを変えるためにはどうすればよいか，調べてみましょう。プログラマーズマニュアルが役に立つはずです。調べてみるとDOSコール，IOCSコールそれぞれにCRTモードを変えるためのファンクションコールがあるのがわかりますね。DOSコールは，

conctrl (67～69pp.)

IOCSコールは，

CRTMOD (167～168pp.)

です。今回はDOSコールでいきましょう。

ところで，このリスト1を打ち込むわけですが，これはエディタで打ち込んだからといってそのまま実行できるわけではありません。冒険に出よう（プログラムを組もう）とする人がたいてい最初に使う言語B

リスト1

```
 1:           *サンプルその1
 2:           *         By Decno
 3:           *        1990/05/22
 4:           .list
 5:           .include DOSCALL.MAC
 6:           .include IOCSCALL.MAC
 7:           .text
 8: _csize              equ     16                *画面サイズの変更モード
 9:
10: *プログラム領域
11: main:           lea      usersp,sp          *スタックの設定
12:                 move.w  crtwok,-(sp)       *CRTモードを設定
13:                 move.w  _csize,-(sp)       *モードの設定
14:                 DOS     CONCTRL            *CRTMOD
15:                 addq.l #4,sp
16:                 DOS     EXIT               *作業終了
17:
18: *データ領域
19:                 .data
20:                 .even
21: crtwok:         .dc.w   5                  *ドットは512*512・65535色
22:                 .ds.b     4096             *スタック
23: usersp:
24:                 .end
```

ASIC。そのBASICのようなインタプリタと，ほかのほとんどの言語との最大の違いは，

エディタがついてない

ということでしょう。BASICの場合には，BASICに入ったらそのまま，

10 int a, b /*変数の宣言*/

というふうに，

行番号　実行する内容＋リターン

とキーを押すことでプログラムを作って，

RUN

で実行していました。

ところが普通一般の言語というのはこの部分が一緒になってはいないわけです。たとえばアセンブラでコマンドライン用プログラムを組む場合，

1)　エディタでプログラムを打ち込む
2)　アセンブラでマシン語のプログラムを作る
3)　コマンドラインでファイル名を打って実行

というふうにステップが分かれているのです。

そこで，さっきの例にも出てきましたがプログラムを打つ部分を受け持つエディタというものが必要になるのです。ここではエディタはED.Xを使うものとして説明をします。

まずはエディタの立ち上げ。エディタを立ち上げるのにはコマンドラインで，

A> ED

とするかVSでエディタのアイコンをダブルクリックするわけですが，コマンドラインではそのとき一緒に打ち込むファイル名を指定できます。

A> ED CRT3.S

このCRT3.Sというプログラム（このエディタで打ち込んだだけでまだ実行できないプログラムのことをソースファイルといいます）が存在していなかったら新規のファイルを，そうでなかったらその前に書いたプログラムの修正などができます。

EDで打ち込むときにはアセンブラのプログラムの場合は拡張子は必ず，

○○○.S

のように.Sにしてください。

EDを使う

さて実際の打ち込みを行うわけですが，打ち込むだけならBASICとも感覚的にはそんなに違いません。次々とリストを（もちろん行番号はつけずに）打ち込んではリターン，打ち込んではリターンを繰り返すだけです。打ち込んだ行の修正は↑↓→←のキーとROLL UP/DOWN・HOMEキーでカーソルを移動して上書き（このときINSキーが押されていると挿入）したりDELキーで文字を削ったりすればいいわけです。

で，プログラムを組むわけですがリスト1には，

```
.list
.include DOSCALL.MAC
.include IOCSCALL.MAC
.text
```

というのがありますね。これはDOSコール，IOCSコールを使うファイルのリストが始まるよ……といういわばアセンブラに対するおまじないです。最初のうちはひとつ覚えてよいでしょう。ちなみに終わりのおまじないは，

```
.end
```

です。

最初と最後のおまじないのあいだにあるプログラムの中身はプログラマーズマニュアルにサンプルが出ていますのでそれを参考にしながら作っていきましょう。

プログラマーズマニュアルにはこのDOSコールのサンプルとしてこんなリストが出ています。

```
move.w  y,-(sp)  *MDにより異なる
move.w  x,-(sp)  *MDにより異なる
move.w  #3,-(sp)
dc.w    _CONCTRL
addq.l  #6,sp
```

これを画面モードの変更用に手直ししてやればいいわけです。

そして，プログラマーズマニュアルの69ページによれば画面モードを変えるには，

MD＝16 conctrl(16, MOD)
MODで指定する画面モードにします。
MODの内容は次のとおりです。

⋮

MOD＝5 高解像度512×512，グラフィック65536色　（以下略）

だそうですから，さっきのサンプルの，

y, x　を　MOD　に
3　を　5　に

変えるだけでよさそうです。つまり，

```
move.w  #5,-(sp)   *mod
move.w  #16,-(sp)  *
dc.w    _CONCTRL
addq.l  #4,sp
```

とすれば画面モードが変わるに違いありません。

気をつけてほしいのはMOVE命令で使っている値の頭に#がついていることです。68000のマシン語では変数の値（つまりメモリ上の値）を使うにはなにもつけないのですが，直接2とか3とか数字を代入する場合には#マークをつけるのです。

実はサンプルの，

move.w　y,-(sp) *MDにより異なる

という行はYという変数の中身（Yが指すアドレスのメモリの持つ値）を代入していたのですね。

さて，ついでですからあとで読みやすくするために数字などのパラメータの部分を細工してしまいます。まず，マニュアルではDOSコールの呼び出し方を，

dc.w　_CONCTRL

としていますが，実はここはさっきのおまじないで，

DOS　_CONCTRL

と書くことができるのです。このほうがDOSコールだということがわかりやすくなるのでよいでしょう。

それから，BASICでたとえば，

conctrl(16,5)

を，

mod＝5
conctrl(16,mod)

と読みやすくするために代入したりすることがありました。これと同じことをMODを設定するのに使ってみましょう。マシン語ではプログラムと変数は分けておかなくてはなりません。このことに注意して考えてみるとさっきのプログラムなら，プログラム部分の，

move.w　#5,-(sp)

を，

move.w　mod,-(sp)

にして，データ部分に，

mod:　.dc.w　5

というのを作ってやればいいわけです。データ部分が始まるおまじないは，

```
.data
.even
```

で，プログラム領域の後ろにつけます。

それから代入に近いものとしてラベルに値を持たせることができます。つまり，たとえば_csizeというラベルを16という数字の代わりに使うことができるようにすることができるのです。そのためのおまじないがリスト中にも出てくるequ疑似命令です。

_csize　equ　16　*

こんなふうに，

ラベル名　equ　値

というふうに書くことになっています。

注意が必要なのはこれは変数ではないためにこのラベルに数値の代入はできないということです。つまりさっきのmodに，

move.w　#1, mod

(BASICでいうmod=1にあたる)はできますが,

move.w　　#1, _csize

ということはできないのです。

さて,あとはプログラムの実行を終わらせてコマンドモードに戻るだけです。プログラムリストの終わりを示す.endはリストの最後であることは伝えられるのですが,それはプログラムの終わりであるとはいっていないのです。プログラム自体はDOSコールで終わらせます。

DOS　_EXIT

がそれです。

さ,これでだいたいプログラムは組めました。これからアセンブルに入ります。

アセンブルして実行だ!

次はアセンブラおよびリンカの使い方です。アセンブラやリンカの起動は,

A>AS　ファイル名

A>LK　ファイル名　ファイル名2……

です。

アセンブルの際にファイル名に拡張子をつけなければ勝手にアセンブラが.Sファイルであると解釈してくれます。したがって普通に作っているうちはファイル名に拡張子をつける必要はありません。

またAS.XとLK.Xにもスイッチがありますが,デフォルトでだいたいこと足りるのであまり使うことはありません。

AS.Xのスイッチ

/P……リスト出力を行う(デフォルトではファイル名.prnに出力)

/W……警告メッセージの出力禁止

LK.Xのスイッチ

/x……シンボルテーブルの出力禁止

まぁ,こんなところでしょう。

では実際にリスト1をアセンブルしてみましょう。リスト1をCRT.Sという名前でエディットしているとしましょう。

このときエディタをいったん終わらせてもいいのですが,せっかくですからエディタからどうにかしてしまいましょう。例によっておまじないがあります。アセンブルのためのおまじないは,

ESC･H

ESC･C

AS　　CRT

ESC･C

LK　　CRT

これでアセンブルは終わりました。しかしなにかメッセージが出てきました。どうやらエラーがあってアセンブルできなかったようです。

```
X68k Assembler v1.01 Copyright 1987 SHARP/Hudson
Warning : Line 13 absolute addressing
move.w _csize, -(sp) *モードの設定
line 14 undefined symbol error
line 16 undefined symbol error
undefined symbol
CONCTRL
EXIT
2 Fatal error(s)
```

なるほど,14行目と16行目にエラーがあったようです。では14行目を見てみましょう。まず,カーソルを14行目にあわせます。カーソルキーで運んでいってもいいですが,もっと簡単に探す方法がED.Xにはあります。ESCキーを押してから14,リターンという順に押してください。カーソルが14行目に飛んだでしょう。

さて,なにが間違えていたのでしょう。そうです。ラベル名の前のアンダーバー("_")を忘れていたのです。IOCSCALL.MACやDOSCALL.MACではファンクションコール名の前に_が必要なのです。CONCTRLは_CONCTRL, EXITは_EXITとしなくてはなりません。

ほかにもなにやら警告が出ていますがエラーではないということでとりあえずほっておきましょう。ということで14,16行を直したのがリスト2です。さ,もう一度アセンブルし直しです。

もう一度ESC･Hからlk crt3までのおまじないを打ち直しましょう。

またしても警告は出ていますが,とりあえずエラーは出ず,.Xファイルができたようです。一度エディタから抜けましょう。

ESC･E

これでエディタから抜けました。.Xファイルができていますのでコマンドモードでファイル名を打てば(あるいはVSモードでマウスをその.Xファイルのアイコンでダブルクリックすれば)実行できます。

A>CRT3

なぜか,バスエラーが出てしまいました。やはり,先ほどの13行のワーニングがまずかったのでしょうか? 先ほどのワーニングの部分,13行をよく見てみましょう。なにかバグの原因になりそうなものがありますか? ありました。

move.w　_csize, -(sp)　*モードの設定

_csizeはラベルに値を持たせたもので変数ではありません。直接,数値を書く(イミディエイトモードといいます。反対に変数(メモリ上の値からとってくる)はエクステンドモードといいます)場合は#マークをつけなくてはいけないのでした。

このように必ずしもエラーメッセージがでないバグというのもマシン語のプログラムには多々あるのです。ではその部分を直したのがリスト3です。さあ,今度こそうまくいくでしょう。エディタに入って書き直して,アセンブルして実行してみましょう。

A>CRT3

どうですか。今度はうまく512ドットモードの画面になったでしょう。やっとマシン語のプログラムが1本完成しました。あ

リスト2

```
 1:          *サンプルその1
 2:          *          By Decno
 3:          *       1990/05/22
 4:          .list
 5:          .include DOSCALL.MAC
 6:          .include IOCSCALL.MAC
 7:          .text
 8: _csize           equ     16                  * 画面サイズの変更モード
 9:
10: *プログラム領域
11: main:          lea      usersp,sp        * スタックの設定
12:                  move.w  crtwok,-(sp)     * CRTモードを設定
13:                  move.w  _csize,-(sp)     * モードの設定
14:                  DOS     _CONCTRL         * CRTMOD
15:                  DOS     _EXIT            * 作業終了
16:
17: *データ領域
18:                  .data
19:                  .even
20: crtwok:          .dc.w   5                *ドットは512*512・65535色
21:                  .ds.b     4096           *スタック
22: usersp:
23:                  .end
```

リスト3

```
 1:          *サンプルその1
 2:          *          By Decno
 3:          *       1990/05/22
 4:          .list
 5:          .include DOSCALL.MAC
 6:          .include IOCSCALL.MAC
 7:          .text
 8: _csize           equ     16                  * 画面サイズの変更モード
 9:
10: *プログラム領域
11: putank:          lea     usersp,sp        * スタックの設定
12:                  move.w  crtwok,-(sp)     * CRTモードを設定
13:                  move.w  #_csize,-(sp)    * モードの設定
14:                  DOS     _CONCTRL         * CRTMOD
15:                  DOS     _EXIT            * 作業終了
16:
17: *データ領域
18:                  .data
19:                  .even
20: crtwok:          .dc.w   5                *ドットは512*512・65535色
21:                  .ds.b     4096           *スタック
22: usersp:
23:                  .end
```

なたのレベルもひとつ上がったようです。

でも，512ドットモードになったはいいが768ドットモードに戻れませんね。そうです，512ドットモードになるプログラムは作ったのですが786ドットモードに戻すプログラムを作っていないのです。

ついでにもう1本戻るプログラムを作ってみてください。なに，MODを5から0にするだけのことです。自分でやればもう一段レベルが上がることでしょう。

冒険の書(その2)

さて，もう1本プログラムを作って，いままで「おまじない」で済ませてしまったところを説明することにしましょう。リスト4はマウスカーソルを好きなパターンに書き直すプログラムです。このプログラムも例によってアセンブルして.X形式にして実行しなければなりません。基本的にはリスト3とまったく同じです。

```
.list
.include DOSCALL.MAC
.include IOCSCALL.MAC
.text
```

で始まって，

```
.end
```

で終わっています。

まず，このあいだに作ったプログラムと違ってこれらは68000MPU自体の命令ではありません。これらは疑似命令といってアセンブラやリンカに対して行う命令なのです。アセンブラマニュアルでは213ページからの解説がそうです。

AS.Xには.dc命令のほかにもたくさんの疑似命令があります。そのなかでも一番よく見かけるのは，

```
.even
```

でしょうか。この命令は次の命令(moveや.dcなど)が偶数番地から始まるようにするものです。これはX68000で使用しているMC68000というMPUが「ワード以上の単位のメモリアクセスは偶数アドレスでしかできない」という制約によるものです。

68000MPUの命令はすべて2バイト以上の長さでかつ2バイト単位に大きくなるので奇数番地からはプログラムを動かすことができない。つまり，プログラムの先頭に必ずつけるようにしないと原因のわからないバグに悩まされることになりかねないからです。

.include命令はほかのマクロファイルを読む命令でサンプルリストでは，

```
DOSCALL.MAC
IOCSCALL.MAC
```

をコールするのに使っています。これによってDOS&IOCSコールの呼び出し方を定義してあるマクロをリンカが呼び出しているのです。マクロの作り方がわかるようになったらぜひこのDOSCALL.MAC，IOCSCALL.MACの中身を見てみるといいでしょう。

equ は先ほども出てきたようにシンボルの値を決めるのに使います。この機能をうまく使うと，たとえば……，

```
extlop:
        movem.l   $e20000,-(sp)
        movem.l   $e20004,-(sp)
        movem.l   $e20008,-(sp)
        movem.l   $e2000c,-(sp)
        movem.l   $e20010,-(sp)
        movem.l   $e20014,-(sp)
        movem.l   $e20018,-(sp)
        movem.l   $e2001c,-(sp)
```

こんなリストを転送元の対象を$E20000から$E40000に変更したいときには，

```
extlop:
        movem.l   $e40000,-(sp)
        movem.l   $e40004,-(sp)
        movem.l   $e40008,-(sp)
        movem.l   $e4000c,-(sp)
        movem.l   $e40010,-(sp)
        movem.l   $e40014,-(sp)
        movem.l   $e40018,-(sp)
        movem.l   $e4001c,-(sp)
```

このように全部書き換えなくてはなりませんが，EQUを使って，

```
_txtvrm   equ     $e20000
extlop:
        movem.l   $_txtvrm,-(sp)
        movem.l   $_txtvrm+4,-(sp)
        movem.l   $_txtvrm+8,-(sp)
        movem.l   $_txtvrm+12,-(sp)
        movem.l   $_txtvrm+16,-(sp)
        movem.l   $_txtvrm+20,-(sp)
        movem.l   $_txtvrm+24,-(sp)
        movem.l   $_txtvrm+28,-(sp)
```

と書いておけば，シンボルの値を変えて，

リスト4

```
==================  4  ==================
     1:            *サンプルその2
     2:            *        By Decno
     3:            *        1990/05/25
     4:            .list
     5:            .include DOSCALL.MAC
     6:            .include IOCSCALL.MAC
     7:            .text
     8: _ms_vpat:          equ      6                  * 表示するマウスカーソルの番号
     9:
    10: *プログラム領域
    11: main:              lea      usersp,sp          * スタックの設定
    12:                    move.l   msno,d1            * マウスカーソルの指定
    13:                    lea      pat1,a1            * マウスカーソルのパターン

    14:                    IOCS     _MS_PATST
    15:                    move.l   #_ms_vpat,d1
    16:                    IOCS     _MS_SEL
    17:                    DOS      _EXIT              * 作業終了
    18:
    19: *データ領域
    20:                    .data
    21:                    .even
    22: msno:              .dc.l    6                  *マウスカーソルの番号は1番
    23: pat1:
    24:                    .dc.w    0,0
    25:                    .dc.w    %1111111111111111
    26:                    .dc.w    %1111111111111111
    27:                    .dc.w    %1110000000000000
    28:                    .dc.w    %1111100000000000
    29:                    .dc.w    %1101110000000000
    30:                    .dc.w    %1100111000000000
    31:                    .dc.w    %1100011100000000
    32:                    .dc.w    %1100001110000000
    33:                    .dc.w    %1100000111000000
    34:                    .dc.w    %1100000011100000
    35:                    .dc.w    %1100000001110000
    36:                    .dc.w    %1100000000111000
    37:                    .dc.w    %1100000000011100
    38:                    .dc.w    %1100000000001110
    39:                    .dc.w    %1100000000000111
    40:                    .dc.w    %1100000000000011
    41:
    42:                    .dc.w    %1111111111111111
    43:                    .dc.w    %1100000000000000
    44:                    .dc.w    %1010000000000000
    45:                    .dc.w    %1001000000000000
    46:                    .dc.w    %1000100000000000
    47:                    .dc.w    %1000010000000000
    48:                    .dc.w    %1000001000000000
    49:                    .dc.w    %1000000100000000
    50:                    .dc.w    %1000000010000000
    51:                    .dc.w    %1000000001000000
    52:                    .dc.w    %1000000000100000
    53:                    .dc.w    %1000000000010000
    54:                    .dc.w    %1000000000001000
    55:                    .dc.w    %1000000000000100
    56:                    .dc.w    %1000000000000010
    57:                    .dc.w    %1000000000000001
    58:
    59: crtwok:            .dc.l    5                  *ドットは512*512・65535色
    60:                    .ds.b      4096             *スタック
    61: usersp:
    62:                    .end
```

```
    _txtvrm    equ     $e40000
```

とすれば変更も簡単です。

サンプルリスト中にはほかにも,

```
    .list
    .text
    .data
```

があります。これにライブラリを作る際に必要な,

```
    global
```

も含めて, この4つはぜひアセンブラマニュアルででも調べてみてください(アセンブラマニュアルを引いてみるいい練習にもなるでしょう)。

ところでこのプログラムにはマウスのパターンがデータ領域に2進数の定数で定義されています。リストを見てみると0と1でなにやら絵が描かれているところがありますね, それがそうです。

しかし, これをそのままリストの上から順に0,1で書き込んでいくとなにやら間違えてしまいそうです。BASICならこんな場合どうしていましたか? 普通はなにか1行作っておいて行番号の部分だけ変えてリターン, 行番号の部分だけ変えてリターン……とやって行のコピーをしたんじゃないかと思います。

これと似たようなエディタの機能として行のカット&ペーストがあります。ESC・P(エスケープを押してからPを押す)でカット, ESC・Gでペーストです。ESC・nPでn行カット, ESC・nGでn回ペーストです。これを使えばこういうときに,

```
    .dc.w   %0000000000000000
```

という行をカット(ESC・P)して32行ペースト(ESC・32G)して, その上に1で絵を書き込んでいく(もちろん上書きモードで)ようにすれば簡単に展開できます。

いま, ESCキーを押して使う機能が出てきましたがED.XではよくESCキーを使用します。子プロセスの起動はESC・C(BASICでいう!コマンドです。ESC・Cとキーを押すとパラメータを聞きにきます),

ESC・Hでファイルをセーブ
ESC・Yでファイルのマージ
ESC・Tでファイル名変更
ESC・Eでセーブして終了
ESC・Qでセーブしないで終了

などがあります。これで先ほどの「アセンブラのおまじない」のタネがわかったことでしょう。

データ長に注意

68000のアセンブラでプログラムを組む際に一番注意を要するのはなんといってもデータ長の問題でしょう。68000ではバイト(8ビット), ワード(16ビット=2バイト), ロングワード(32ビット=4バイト)という長さのデータを一度に扱うことができるわけですがそれぞれデータを扱う際にデータ長を指定しなくてはならないのです。

アセンブラでなんらかのテーブルなどを作るときにメモリ上にデータを置きたいなどということがあります。それを実現するのがアセンブラ疑似命令のひとつ,

```
    .dc
```

なのですが, このときもデータ長を指定しなければなりません。たとえば, ワードで100(16進数)という数字を置いておきたいときは,

```
    .dc.w        $100
```

と書かなくてはなりません。

またバイト長データで10進数の100のときは,

```
    .dc.b      100
```

とデータ長を.bにし, $を取り去ります。

ちなみにロングワードで100(2進数)なら,

```
    .dc.l          %100
```

になります。

実は先ほどのマウスカーソルのパターンも16進数で32個数字を並べてやっても構わないし, 実際, 単なる数値として扱われているのですが, 2進数で描くとうまく絵のように見えるのでそのようになっているのです。

(で)の置き手紙……

さて, 勇者よ。2つの冒険を経て君はもうひとりで立派にマシン語に挑むことのできる戦士として十分に経験を積んだことと思う。これからしばらく別々に行動しよう。ここから君が冒険に出るも, 家に帰るも君しだい, というわけだ。しかしながら勇者よ, できればもっと成長した君とこのマシン語の世界でもう一度会いたいものだ。

BASICしかやってない君でもツールやマシン語プログラムのコツを一度覚えてしまうと意外に簡単なもんだと感じたことだろう。それにひとつマシン語の世界を知っていればほかのマシンのマシン語の世界で冒険するのもそう難しいものではないし, またCなどに行こうと考えたときにも中身がなにをしているかがわかると非常にやりやすいだろうと思う。がんばってX68000でマシン語をマスターしてくれたまえ。幸運を祈っている……。

DB.Xは最後の武器だ!

マシン語のデバッグに欠かすことのできないものにデバッガ, db.xがあります。ではマシン語のデバッグではどんなことをするのか? それはデバッガがどういう機能を持っているか知ることでだいたいわかるのではないかと思います。db.xの主な機能は次のとおりです。

- **1行アセンブル, 逆アセンブル機能**
- **ブレイクポインタ機能**
- **メモリ操作**
- **レジスタ操作**

デバッガもコマンドラインで起動しますが,

```
    A>DB  ¥BASIC2¥BASIC.X
```

というふうにしっかりと,

```
    ファイル名.拡張子
```

と指定しなければなりません。また当然のようにエディタからも,

```
    [ESC]C
    DB ファイル名
```

で起動できます。

とりあえず,

```
    -G
```

でプログラムを実行します。

単なるバスエラーあたりだとこのコマンドで実行してみて変なアドレスをアクセスしてないか確認するだけでも結構効果があります。実際, リスト2をデバッガにかけてみると,

```
    -g
    Exceptional Abort By bus error
    By Memory Access of 00000010
    at 0008668C move.w $00000010, -(A7)
```

と出てきて「確か16という数字は画面モードの変更をするときのパラメータだった→それがアドレスとして処理されている→イド抜けだ」というふうにすぐわかるのです。

あとデバッグの鉄則というのに「他人の目で見よ」というのがあって, 早い話が自分で作ったプログラムを自分のものでないような気で見れば意外とバグが見つけやすいのですが, そのとき役に立つのが,

```
    -L  アドレス
```

です。

これはアセンブラがソースから実行ファイルに変換したものを逆に実行ファイルをソースに戻す(逆アセンブル)ということをしてくれるのです。はたして, 逆アセンブルなんかが役に立つのだろうかというと, これははっきりいってかなりの効果です。

ほかに68000でよくあるバグは次のとおり。

even抜け……ワークエリア中にバイト長のデータがあってその後ろのワード長データをアクセスする際にアドレスエラーになることが多い。Xコマンドでレジスタをよく見るとよい。

イド抜け……イミディエイトアドレッシング(#$E28000)の際に#マークを忘れること。#マークが抜けてもエクステンドと解釈されるので見つけにくいことがある(特にアセンブラが警告を出さないようにした場合)。

DOSコール&IOCSコールを使う

Kageyama Hiroaki
影山 裕昭

X68000でマシン語プログラミングを始めるには，まずX68000に用意されているDOSコールとIOCSコールを使ってみるのがよいでしょう。ここではサンプルを交えて各コールの使い方を中心に解説していきます。

今年も春になってX68000に新たに3機種の新製品が発表されました。高校，大学進学のどさくさにまぎれて親のスネをかじる人，あらかじめ予想して冬のうちからお金を貯めてた人，もうとっくに新製品を購入してしまった人もたくさんいることでしょう。

なんにしても共通にいえることは，コンピュータを目的もなく買う人はいないということです。さて，あなたはX68000になにを求めているのでしょうか。ゲーム，レイトレーシング，MIDI，やろうと思えばなんだってできてしまうところが大きな魅力ですが，志半ばにしてすでに高価なゲームマシンとなりつつある人も少なからずいるに違いありません。

レイトレにしてもMIDIにしても市販されているソフトを利用することは時間的にも技術的にも大きなメリットがあります。しかし，自分でコンピュータを操作しているという感覚に欠けることはいなめません。犬を飼うときも犬小屋を買ってくるのは簡単ですが，鋸とトンカチを使って自作して完成した犬小屋は，たとえ格好が不細工であっても，そのときの満足感はお金で味わえない貴重なものなのです。

ここでは皆さんにプログラムを作る醍醐味をわかってもらうためのアプローチとして，MC68000のアセンブラを使うときに最低限知っておいてほしいことをお話ししましょう。初心者の方にわかりやすく説明していくつもりなので，いままでアセンブラを使ったことのない人もぜひ最後までおつきあいください。

DOSコールとIOCSコール

マシン語はなんでもできる言語（?）です。ハードウェアを直接操作すれば，BASICやC言語で不満のある速度しか出なかった処理も高速実行可能です。しかし，このことを裏返せば，マシン語はなにもかもやらなければならない言語でもあります。本来なら，画面にキーボードから打った文字を表示するだけでも，アセンブラの知識はもちろん，数多くのコントローラやサブCPUなどのハードウェアに通暁していなければなりません。

さらに「割り込みが発生したら」「エラーが起きたら」という不慮の事態に備えておかないととても安心してプログラムを実行できません。これもかなり面倒な処理です。

これではたまらないので，BIOSやOSというものが存在しているわけです。これらはシステムを管理すると同時に膨大なハードウェアを有効に使うためのさまざまなサービスを提供しています。

DOSコールとはHuman68kで使われているファイル入出力などの複雑な処理を，ユーザーが簡単に扱えるようにとOS側が提供しているものです。主な機能として，

- キー入力
- コンソール表示
- プリンタ，RS-232C入出力
- ファイル管理
- かな漢字変換制御
- ファイルの起動，プロセス管理

などがあります。

DOSコールはOSと密着しているものなので，OSのリダイレクト機能を利用することも，もちろんできます。また，Human68kを使う限りプログラムの常駐，終了をするときもDOSコールを利用することになります。

DOSコールの利用には未定義命令であるラインFのエミュレータを使い，パラメータを必要とする場合はスタックを介してやりとりが行われます。見かけ上68000のマシン語にDOS管理用の命令が加わったような感じで使うことができます。

アセンブラでは次のように表されます。

```
dc.w $FFxx
```

xxには00～FF_Hが入りますが，プログラマーズマニュアルを見てもわかるとおり，ところどころ数字が飛んでいます。これはDOSコールが将来拡張されるときのことを考えているのであって，実際にHuman v2.01では，初期のものに比べていくつかのDOSコールが追加されています。

他方，IOCSコールはX68000がROMで持っている基本入出力サブルーチンを利用するもので，X68000の持つすべての機能を使うためのサービスが用意されています。各種コントローラとのやりとりからマウス，グラフィック，割り込み制御までサポートされています。

DOSコールがOSと密着しているのに対して，IOCSコールはOSから独立しているのでどんな状態からでも使えます。

IOCSコールを使うには，レジスタD0にコール番号をセットして，MC68000のTRAP #15を使い，必要なパラメータの受け渡しには指定されたレジスタを使います。

```
moveq.l #$xx,d0
trap    #15
```

(xxはIOCSコール番号)

DOSコールとIOCSコールがどういうものであるか少しはわかっていただけたでしょうか?

開発の手順

X68000によるアセンブラプログラムの開発は，

1) プログラムの作成，入力
2) アセンブル
3) リンク
4) 実行

の手順を経て行われます。実際には1回で仕様どおりの結果が得られることは稀なので，さらにデバッグという作業を経て，1)から繰り返すことになります。それでは最初はDOSコールを使ったプログラムの入力から実行まで，順を追って説明していくことにしましょう。

●入力

アセンブラで書かれたプログラムの入力にはエディタを使います。カレントドライブをBにして，

B>A：ED SAMPLE1.S

のように，これからSAMPLE1.Sというファイルを Bドライブに作成することを命令します。

早速リスト1を入力してみましょう。カーソルの移動や文字の削除などの基本的な編集操作は，BASICのプログラムを入力する場合とほとんど同じですから，難しいことはなにもないはずです。

リスト中の＊はプログラムに注釈をつけたいときなどに使うもので，＊以降の文はアセンブルするときに無視されるものです。すべて入力し終わったら，ESCに続けてEと入力してください（以降ESC・Eのように表現する）。これでBドライブにSAMPLE1.Sが保存されました。

リスト1のようなアセンブリ言語で記述されたプログラムをソースリストと呼びます。ソース(source)とは「源の」という意味の英語からきているものですが，知らなかった人は覚えておいてくださいね。

●アセンブル

ソースリストを保存したら次の作業はアセンブルです。これにはアセンブラ（AS.X)を使います。アセンブラの役割はアセンブリ言語で記述されたソースリストを，CPUが直接わかるかたちである2進数（16進コード）に変換したり，メモリ領域を確保したりすることです。

B>A：AS SAMPLE1.S

としましょう。拡張子が.Sであれば，

B>A：AS SAMPLE1

でもかまいません。正常に終了すれば，

No Fatal errors(s)

と表示され，拡張子をOに変えたSAMPLE1.Oというファイルが新たにカレントドライブに作られます。

この拡張子がOのファイルをオブジェクトファイルと呼びます。先ほどカレントドライブをBに変更したのは，オブジェクトファイルをドライブ1に作成したかったからなのです。

さて，エラーがあった場合はエラーの発生した行番号と，エラーの種類が表示されますので，それをメモして再びエディタを起動して間違いを訂正して再アセンブルします。

●リンク

一般に数千行から数万行にも及ぶ大規模なプログラムを開発する場合などは，プログラムをその機能ごとにモジュールに分割して作成していくのがごく当たり前となっています。これらモジュールごとに分割され作成されたオブジェクトファイルをまとめあげ，ひとつの実行可能ファイルに作りあげるのがリンカの役割です。

特にC言語では関数のほとんどがライブラリのかたちで提供されており，リンカはライブラリの中から必要なオブジェクトファイルを取り出すという重要な働きをしています。

ところが，このサンプルプログラムのように自己完結しているプログラムならリンカを通す必要はないと感じるかもしれませんが，そこはそれ。リンカを通さないと実行可能ファイルが作成されないことになっていますので，「アセンブルしたらリンカを通す」と覚えておいて間違いはありません。ではリンクしてみましょう。

B>A：LK SAMPLE1.O

アセンブルのときと同じように，拡張子がOであれば，拡張子を省略することができます。無事リンクが終了すると，拡張子がXになった，SAMPLE1.Xという実行可能ファイルがカレントドライブに作成されます。実行可能ファイル名を変えたいときはオプションを指定して，

B> A：LK /OA：TEST SAMPLE1

のようにします。こうすればTEST.XがAドライブに作られます。

●実行

実行は単純に，

B>SAMPLE1

とするだけです。このプログラムは画面に文字列を表示するものだったのです。まあ，ソースリストを見て容易に想像はついていたでしょうが。

いままで説明してきたような手順をふんで，やっと実行可能なファイルを作成することができるのです。

プログラムの解説

さて実行できたところで，とりざたされていたプログラムの説明をすることにしましょう。

3，4行目にあるのは疑似命令と呼ばれるものでMC68000にある命令ではなく，アセンブラが使い勝手をよくするために用意している命令です。dcやds命令などを除けば，疑似命令はオブジェクトコードに変換されることはありません。具体的には，.textはプログラムの始まりを指示するもので.evenはアドレスを偶数番地にロケー

リスト1

```
==================  SAMPLE1.S  ==================
   1: *        sample1
   2:
   3:          .text
   4:          .even
   5:
   6:          move.l  #mes1,-(sp)      * 文字列の先頭アドレスを
   7:                                   *   スタックに積む
   8:          dc.w    $ff09            * _PRINT
   9:          addq.l  #4,sp            * スタックポインタを
  10:                                   *   補正する
  11:          dc.w    $ff00            * _EXIT
  12:
  13:          .data
  14:
  15: mes1:    dc.b    'SHARP X68000',13,10,0
  16:
  17:          .end
```

リストファイルの作成

大きなプログラムを開発していると，メモできないほどたくさんのエラーが発生することがあります。そんなときはアセンブラのスイッチを指定すれば，リストファイル（アセンブラの出力するリスト）をディスクに作成することもできます。

B> A：AS /P SAMPLE1

とすれば，SAMPLE1.PRNというリストファイルがカレントドライブに作成されます。

リストファイルをうまく使うには，ED.XでSAMPLE1.Sを読み込んでおいて，ESC・Fと押します。

編集ファイル：

と表示されましたね。ED.Xは同時に10個までのファイルを編集することができるので，これに対して，

編集ファイル：SAMPLE1.PRN

と入力します。すると，リストファイルが画面に表示されました。ここでエラーメッセージのある行を確認したら，ESC・A（SHIFT＋F6）で編集テキストを切り替えて，ソースリストのエラーを訂正しましょう。訂正したら再び編集テキストを切り替えてリストファイルを表示して，ほかのエラー箇所を確認し，同様の作業をエラーがなくなるまで繰り返します。

似たようなものにタグジャンプというかなり便利なものがあるのですが，この機能は残念ながらAS.Xの出力するリストファイルに対しては使用することができないので，宝の持ちぐされといったところです。

タグジャンプ機能を活用した効率的な作業には1990年6月号で村田氏が作成したようなアセンブルドライバを使用することをおすすめします。

ションするものです。.evenが必要な理由はMC68000が奇数番地にワードをアクセスしようとすると，アドレスエラーを発生してしまうことに起因しています。この2つの疑似命令はソースリストで決まって使われるものなので，覚えておいてほしいものです。

6行目からMC68000のアセンブリ言語を使ったプログラムが始まります。このプログラムは文字列表示にDOSコール$FF09の_PRINTを使っていますが(8行目)，最初にいったようにDOSコールでは引き数の受け渡しにスタックを使うことになっているので，引き数をスタックに積まなくてはなりません。この作業をしているのが6行目の，

move.l #mes1,－(sp)

です。#mes1の「#」はイミディエイトデータを指定することを表します。イミディエイト（immediate）とは「直接」という意味の英語で，単なる数値データを指します。

この場合はmes1がイミディエイトデータであることを表しています。mes1はプログラマが作成したラベルです。ラベルはプログラムでサブルーチンを作った場合や，このプログラムのように特定のデータを指し示す場合などにつけるもので，一般的には1カラム目から書き始め，最後にコロンをつけます（15行目）。

ラベル名をつける場合はそのサブルーチンやデータに関連性のある名前をつけるのがよいとされていて，たとえばd0レジスタの値個だけスペースを表示するサブルーチンだったら，

prt_spc_d0:

のようにします。ラベル名にはプログラマのセンスが出てくるので，慣れていない人は本誌に掲載されているソースリストからラベル名のつけ方の「センス」を勉強するといいと思います。

続いてカンマの後ろの－(sp)は，スタックポインタの値を減じることを意味します。いくつ減じるかは，move.?の?の部分によって決定され，b（バイト）なら1，w（ワード）なら2，l（ロングワード）なら4となります。逆に＋(sp)なら増やすことになります。

結局，この命令が実行されると，スタックポインタを4減らし，そこにはmes1の置かれているアドレスが入ることになるのです。こうしてDOSコールを呼び出したあともスタックポインタは4減らされたままですから，スタックポインタに4を足してつじつまを合わす作業が必要となります（9行目）。この一連のスタックの変化を図に表しておいたので参考にしてください。

11行目にあるDOSコールはプログラムの終了に必要なもので，サブルーチンの終わりをrtsで表すのに対して，プログラムの実行を終了するには必ずこのDOSコールを使います。

インクルードファイルを使う

次にリスト1と同じプログラムですが，DOSコールのファンクション番号をシンボルに定義したものがリスト2です。3，4行目のequがシンボルを定義するための疑似命令で，

ラベル： equ 式

で，以降ラベルの値が式の値と同値のシンボルとして扱うことが可能になります。

さらに変更を加えたのがリスト3です。リスト2でシンボル定義だった部分が，

.include doscall.mac

となっています。これはインクルードファイルとしてdoscall.macを使うことをいっているのですが，インクルードファイルってなんだ？ と思う人もいるかもしれません。インクルードファイルとはアセンブラがソースファイルをアセンブルするときに挿入するファイルのことです。挿入するといってもアセンブルが終了すると除外されるものなので，ソースファイルの大きさがアセンブルの前後で変わるようなことはありません。

ここで使っているdoscall.macはCコンパイラのシステムディスクのincludeというディレクトリに収められているもので，すべてのDOSコールのファンクション番号がリスト2と同じ要領でシンボル定義されているファイルなのです。だから，いちいちリスト2のようなシンボル定義しなくてもいいんですね。利用できるものはどんどん利用しましょう。同様に2つのDOSコールの表記方法が，

DOS <シンボル>

となっていますが，このDOSもdoscall.macで定義されているマクロ命令です。マクロとは1行から数行にわたる命令を1行で表せるようにと考えられたもので，doscall.macの中には，

```
DOS     macro   callname
        dc.w    callname
        endm
```

とマクロ命令が定義されているのです。マクロはプログラムが見やすくなるという半面，サブルーチンと違ってアセンブルのときその都度展開されるものなので，あまり多用するといたずらに実行ファイルを大きくしてしまうこともあり，注意が必要です。

インクルードファイルを使うときの注意点は，ソースファイルのあるディレクトリと同じディレクトリにインクルードしようとするファイルがないとエラーになってしまうので，違うディレクトリにある場合はアセンブルするときに，

B> A：AS /Iinclude SAMPLE3

と，スイッチでインクルードファイルのあるディレクトリを指定します。

このプログラムではさらにいくつかの変更点があります。まず，8行目がmove命令ではなくpea命令に置き換わっていますが，やっていることは同じです。一般的にアドレスをスタックに積むときはこちらの命令を使うことが多いようです。10行目もやっ

リスト2

```
==================  SAMPLE2.S  ==================
     1: *        sample2
     2:
     3: _EXIT:            equ     $ff00
     4: _PRINT:           equ     $ff09
     5:
     6:          .text
     7:          .even
     8:
     9:          move.l  #mes1,-(sp)
    10:          dc.w    _PRINT
    11:          addq.l  #4,sp
    12:          dc.w    _EXIT
    13:
    14:          .data
    15:
    16: mes1:    dc.b    'SHARP X68000',13,10,0
    17:
    18:          .end
```

リスト3

```
==================  SAMPLE3.S  ==================
     1: *        sample3
     2:
     3:          .include        doscall.mac
     4:
     5:          .text
     6:          .even
     7:
     8:          pea.l   mes1
     9:          DOS     _PRINT  *
    10:          lea.l   4(sp),sp
    11:          DOS     _EXIT   *
    12:
    13:          .data
    14:
    15: mes1:    dc.b    'SHARP X68000',13,10,0
    16:
    17:          .end
```

ていることは同じですが，addq命令では1から8の加算しかできないので，ほかのやり方も知っておいてほしいのであえて紹介しました。

＊　　＊　　＊

ここでもう一度DOSコールを使う際の注意点をまとめておきましょう。

1)　DOSコールが引数を必要とする場合はスタックに引数を積みます。

```
movem.l d0,−(sp)
pea.l   LABEL
```

などが一般的な手法です。

2)　DOSコールを呼び出します。

```
dc.w $FF09
```

のようにします。

3)　スタックに積んだ分だけスタックポインタの値を増やします。

```
addq.l #4,sp
lea.l  10(sp),sp
```

IOCSコールを使う

いままで作ってきたプログラムをIOCSコールを使って書いてみたのがリスト4です。入力から実行までの手順はDOSコールの場合と同じですから，もう大丈夫ですよね。見てもらえばわかるように，IOCSコールで引数が必要な場合はレジスタに値を入れることになっています。このプログラムでは文字列表示のためにIOCSコール$21の＿B＿PRINTを使いました(9,10行目)。

引数には表示したい文字列が格納されている先頭アドレスをA1レジスタに入れることになっているので，8行目でそのようにセットしています。プログラムの終了にDOSコール＿EXITを使うのに変わりはありません。

リスト6ではインクルードファイルとしてiocscall.macを使ってみたものですが，IOCSコールの呼び出し方が，

```
IOCS _B_PRINT
```

と，リスト4よりもすっきりしているのがわかるでしょう。これもiocscall.macの中に用意されているマクロ命令を使っているからこそできることなのです。

いままで紹介したプログラムは単にDOSコールとIOCSコールの使い方を説明するために作ったプログラムなので，実行してもあまり面白みがありませんでした。そこで，最後に実用的かどうかは別にしても，IOCSコールだからこそできるプログラムを紹介しましょう。リスト6を入力してアセンブル，リンクしたあとに実行すると，あなたのX68000のROMの作成年月日とバージョンを表示してくれます。

使っているIOCSコールは4種類，ただひとつ使われているDOSコール＿EXITはもうお馴染みですよね。プログラムは力ずくで作ったものなので，解析はおすすめできませんが，IOCSコール＿B＿PRINTはエスケープシーケンスコード（45行あたりに見えている）も扱えるんだよ，ということでバージョン表示をカラー文字にしてみました（趣味悪いかなー)。

今回使ったDOSコールやIOCSコールの個々の説明はCコンパイラなどに付属のプログラマーズマニュアルに詳しく書かれていますからそちらを参照してください。

これでひと通りDOSコールとIOCSコールの使い方はわかってもらえたと思いますが，文章だけではまだまだわからない部分もたくさんあると思うので，ぜひとも実際にサンプルプログラムを入力してください。ある程度アセンブラ言語に慣れてきたら，今度は自分でプログラムを作ってみる番です。初めからすんなりいくとは思いませんが，己に負けず頑張ってください。期待しています。

リスト4

```
==================  SAMPLE4.S  ==================
     1: * sample4
     2:
     3:          .include         doscall.mac
     4:
     5:          .text
     6:          .even
     7:
     8:          lea.l    mes1,a1
     9:          moveq.l  #$21,d0          * B_PRINT
    10:          trap     #15
    11:
    12:          DOS      _EXIT
    13:
    14:          .data
    15:
    16: mes1:    dc.b     'SHARP X68000',13,10,0
    17:
    18:          .end
```

リスト5

```
==================  SAMPLE5.S  ==================
     1: * sample5
     2:
     3:          .include         doscall.mac
     4:          .include         iocscall.mac
     5:
     6:          .text
     7:          .even
     8:
     9:          lea.l    mes1,a1
    10:          IOCS     _B_PRINT
    11:
    12:          DOS      _EXIT
    13:
    14:          .data
    15:
    16: mes1:    dc.b     'SHARP X68000',13,10,0
    17:
    18:          .end
```

リスト6

```
==================  ROM.S  ==================
     1: * ROM.S
     2:
     3:          .include         doscall.mac
     4:          .include         iocscall.mac
     5:
     6: ESC:     equu             $1b
     7:
     8:          .text
     9:          .even
    10:
    11:          IOCS     _ROMVER              * ＲＯＭバージョン
    12:          move.l   d0,d1                *   と作成年月日
    13:          move.l   d0,d2                *   を得る
    14:          andi.l   #$00_ff_ff_ff,d1
    15: *
    16:          IOCS     _DATEBIN             * ＢＣＤ記述をバイナリ
    17:          move.l   d0,d1                *   記述に変換
    18:          subi.l   #$0050_0000,d1       * 西暦のオフセット
    19:          lea.l    date,a1              *   ８０を引いて
    20:          IOCS     _DATEASC             * ＢＣＤ記述を文字列に
    21:          lea.l    date,a1              *   変換する
    22:          IOCS     _B_PRINT             * 作成年月日を表示
    23: *
    24:          lea.l    mes1,a1
    25:          IOCS     _B_PRINT             *「バージョン」表示
    26:          swap     d2
    27:          move.w   d2,d3                * ＲＯＭバージョンの
    28:          lsr.w    #4,d3                *   整数部を文字列にして
    29:          move.b   #'.',d3              *   ピリオドを
    30:          addi.w   #$3000,d3            *   くっつけて
    31:          move.w   d3,ver               *   バッファに保存する
    32:
    33:          andi.w   #$0f00,d2            * 少数部を文字列にして
    34:          addi.w   #$300d,d2            *
    35:          move.w   d2,ver+2             * バッファに保存する
    36:          lea.l    attrib,a1
    37:          IOCS     _B_PRINT             * ＲＯＭバージョンを表示
    38: *
    39:          DOS      _EXIT                * プログラムの終了
    40:
    41:          .data
    42:
    43: mes1:    dc.b     ' ＲＯＭバージョン ',0
    44:
    45: attrib:  dc.b     ESC,'[36m'
    46: ver:     dc.b     0,0,0,13,10
    47:          dc.b     ESC,'[33m',0
    48: date:    ds.b     12
    49:
    50:          .end
```

マルチタスクへの挑戦

Kuwano Masahiko
桒野 雅彦

少々高性能なコンパイラが現れても，どうしてもアセンブラを使わないとできないことというものもあります。ここでは，その代表例である割り込み処理を例にマシン語の醍醐味を味わってみましょう。題材は「IOCSコールのマルチタスク実行」です。

Cだって万能ではない

低水準高級言語，あるいは高級アセンブラともいわれるC言語の普及により，アセンブラでゴリゴリとプログラムを作るのは処理速度を極めるような場合など，かなり特殊な場面だけになってしまいました。特にMC68000はミニコンのアーキテクチャを参考に作られたというだけあって，Cからアセンブラに比較的素直に変換されてしまうため，下手なプログラマがアセンブラで書くよりもCコンパイラでコンパイルしたほうがよほど綺麗で高速なコードが出るというようなことも珍しくありません。

システム資源のすべてがメモリ空間におかれる68000では，ポインタ変数さえ使えばどこでも直接アクセスできますから，アセンブラでできることはほとんどCでもできてしまうのです。しかも，制御構造の読みやすさや改造のやりやすさなどではアセンブラの比ではありません。

こうなってくると，いよいよアセンブラでプログラムを書く意義が薄れてしまいます。実際，私の場合も本当に処理速度上問題になるようなところやどうしてもアセンブラでなくては書けないような部分以外はすべてCになってしまっています。

このような状況でCでできるようなことをアセンブラで改めてやってみてもあまり面白くありません。今回はわずかに残されたアセンブラでしかできないことのひとつ，割り込み処理をとりあげ，簡単な時分割処理を行うプログラムを作ってみることにしました。

割り込み動作は個々のCPUに大きく依存するため，高級言語化がもっとも難しい部分でもあります。一部の8086用のCコンパイラやインテル純正(?)言語であるPL/Mなどでは，割り込み処理関数/サブルーチンの記述もできるようにしているようですが，やはりかなり無理があるようです。システムのほとんどがC言語で記述されたOSであるUNIXでも，割り込み処理の入り口などはアセンブラが使われていることから見ても，おそらくこの部分だけは当分のあいだ，アセンブラで記述され続けることでしょう。

割り込み処理

割り込み処理に入る前に割り込みの基本的な考え方について述べておきましょう。

CPUは基本的にメモリ上にある命令をひとつずつ読みとっては実行するだけの機能しかもっていません。いい換えれば，プログラムに書いてなければ，外部でなにがあろうと，一切応答しないわけです。コンピュータがいまでいうところの電卓，いわゆる「計算機」として用いられていたときにはこれでよかったのですが，より高度なシステムの一部として取り入れられるようになり，さまざまな周辺機器が増設されてインテリジェントコントローラとしても用いられるようになってくると，プログラム実行中でも外部からの信号にすばやく応答してもらいたい場面が出てきます。

このようなものに対応するためにプログラムのいたるところでこれらの信号をチェックするように作ることは不可能ではありませんが，プログラムを組むのは非常にやっかいになるうえ，ほとんどの場合は無用であるステータスのチェックに時間をとられるのですから，本来処理させておきたいプログラムの実行速度は低下，CPUの使用効率は著しく悪くなってしまいます。

X68000を例に取れば，キーボード入力やFM音源などがわかりやすいでしょうか。これらの要求速度というのは，CPUの実行速度からすれば，恐ろしく遅いものです。ワープロコンテストの入賞者で10分で千数百文字くらいというのを聞いたことがありますが，仮に文字として計算しても，1秒あたりたかだか3文字，入力文字数としても1秒あたり10文字にしかすぎません。

CPUのほうはといえば，2MHzのZ80でも1秒に10万個程度の命令は処理できることを考えれば，いたるところでキー入力待ちをするというのがCPUにとっていかに無駄な作業かわかることでしょう。

このように非常にのんびりした要求でありながら，その応答速度はわりと厳しいのです。1秒に3回程度しか入力されないからといって，1秒に数回程度しかチェックしないと叩いてから一瞬待って，ようやく入力が受け付けられるという，耐え難いような応答になってしまうでしょう。FM音源なら時間がきたらすばやくレジスタの書き換えを行わないと妙な演奏になってしまいます。

このような要求に対応するために生み出されたのが，「割り込み」という考え方です。CPUに割り込み入力信号というものを付けておき，ここから割り込み要求を伝えると，CPUは現在行っている処理を一時中断し，あらかじめ用意された割り込み処理プログラムの実行を開始します。そして，処理が終わったところで特殊な命令(68000ならRTE命令)を実行すると，再び先ほどまで実行していたプログラムの実行を継続するというからくりです。割り込みが入ったときに，割り込み処理からちゃんと帰ってこられるために必要な情報をセーブする作業はCPUが自動的に行いますので，プログラムを組む側で本来の仕事がどこまで実行されていたかを記録するようなことは必要ありません。

割り込みを使えば，外部からの要求がない限り本来のプログラムの実行を全力で行えますし，要求があれば現在実行している作業が終わりしだい，ただちに割り込み処理プログラムが実行され始めますので応答も非常によくなります。

図1にキー入力を例に割り込みを使わない場合と使った場合のプログラムの動きの違いを示してみました。割り込みを使わない場合には前回のキー入力チェックから次のキー入力チェックまでの期間はキー入力があったとしてもそれに気づくことができ

ませんから，応答をよくするためにはソフトウェアでこまめにキー入力チェックルーチンを呼び出さなければなりません。これが（a）に相当します。

この例ではキー入力チェックにかかる時間1に対して，本来のプログラムが3の時間がたつたびにキー入力チェックを行っています。このため，CPUの処理速度は本来発揮できる速度の3/4になったのと同じことになってしまっています。

これに対し，キー入力を割り込みで行った場合が（b）です。この場合，キー入力がない限り，CPUは本来の処理を全速力で実行できます。しかも，キー入力があってからの応答は（a）の場合よりもよくなっているのです。

CPUの使用効率を下げずに外部からの要求にも的確に応答できるようになる割り込みという考え方はコンピュータにおける最大の発明ではないかと思います。

割り込みとマルチタスク

割り込みが入ると，それまで実行されていたプログラムの実行が強制的に中断されるということを積極的に使い，複数のプログラムを平行して動かすようにしたのが，時分割（タイムシェアリング）処理という手法です。割り込みを一定周期でかけるようなハードウェア（システムタイマとも呼ばれます）を用意して，この割り込みが入ったときにCPUが自動的にセーブした帰り先の情報などをこっそり別の場所に保存し，あらかじめ保存してあった情報と差し替えてなにごともなかったふりをして割り込み処理終了の命令を実行すれば，CPUはなにも知らず，別のプログラムの続きを実行するわけです。これにより，複数のプログラムがシステムタイマが入るたびに少しずつ実行されていくことになります。

この考え方を図2にしてみました。（a）のほうは，マルチタスクの考え方などとしてよく見かける図ですが，これではどうしてプログラムの実行が切り替えられるのかよくわからないので，もう少し正確に書いたのが（b）です。

割り込みが入ると，当然のように割り込み処理プログラムに移行するのですが，ここで帰り先を細工してやれば，RTE命令で戻るのはプログラムBのところになるわけです。CPUはそんなこととは露知らず，プログラムBのプログラムを実行していきます。そのうちまたタイマ割り込みが入り，今度はプログラムCを実行，次のタイマ割り込みでは最初に保留されていたプログラムAに戻って，先ほどまでの続きを実行にかかるという具合になるわけです。

このような時分割の考え方をさらに一歩進め，並行して動くプログラム同士での情報交換や，あるプログラムからほかのプログラムの起動，ディスクやプリンタのように複数のプログラム同士のあいだで共有しなくてはならない資源の管理などのサポートを行うようにしていったのが，マルチタスクOSと呼ばれるものです。時分割で実行されるプログラム1つひとつをタスク（task：仕事，作業）と呼んでおり，これらが同時に複数動くのでマルチタスクと呼ぶわけです。

パソコンの世界ではOSというとMS-DO

図1　割り込みを使った場合と使わない場合の違い

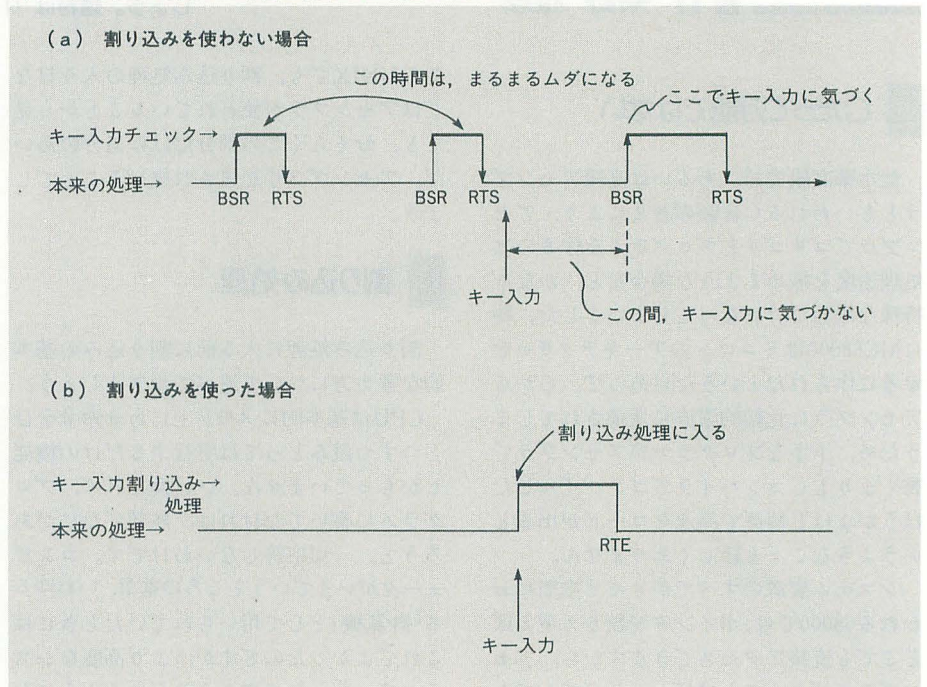
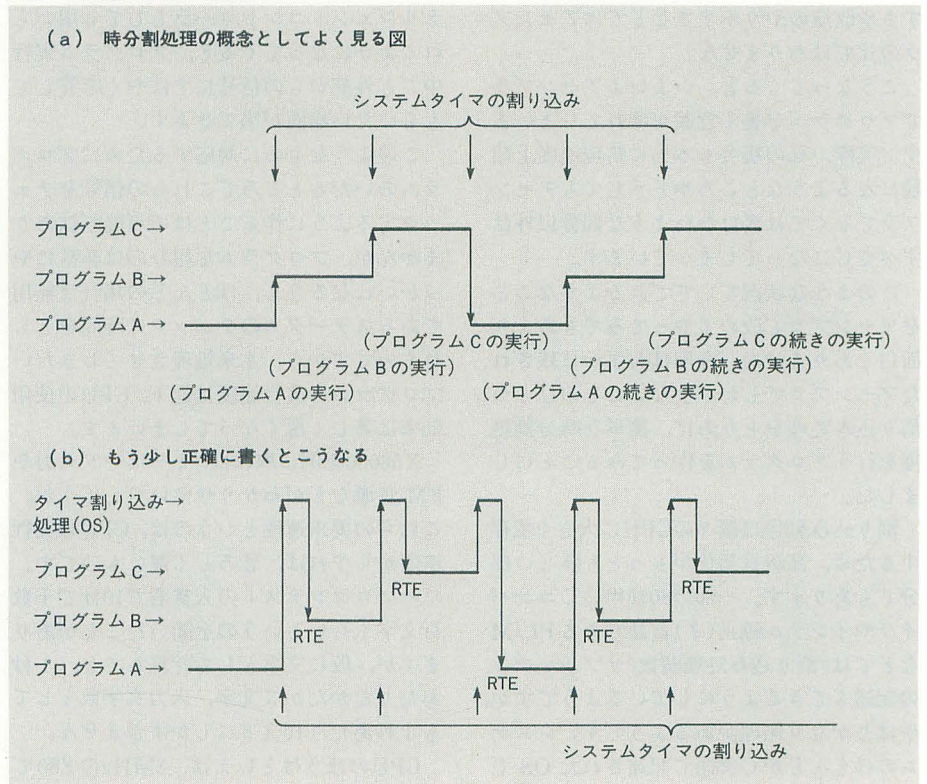



図2　時分割処理の考え方

Sが主流のようですが，これらは複数のタスクを動かす機能のない，シングルタスクのOSです。マルチタスクになるとシングルタスクのときには考えなくてよかったような，複数のプログラム同士の競合やシステムの持つメモリや入出力デバイスの取り合いなどの管理に加え，タスク同士の通信などのサポートも必要になるためOS自体の構造が複雑になるのですが，マルチタスク構造をうまく使ってやるとシングルタスクではやっかいだったことが大した苦労もなく動かせたり，効率のよい動きをさせることができるようになります。

たとえば，シングルタスクのOSでファイルの中身をプリンタに打ち出すようにするとしましょう，プリンタに打ち出すプログラムは，ファイルを読み出してはプリンタに送りつけるというだけの単純なループですが，このあいだほかの作業はまったくできなくなります。これをいくらかでも回避するためにプリンタスプーラなどのソフトが作られてPDSでもずいぶん出回っていますが，独自のバッファを用意して，プリンタからの割り込みを横取りして……など，かなりの小細工をしており，単純にファイルを読み出して打ち出すプログラムとは似ても似つかないような内容です。

これがマルチタスクOSの下だったらどうでしょう。話は簡単です。ファイルを読み込んでプリンタに送るプログラムを新規のタスクとして起動してしまえばいいのです。

たとえばUNIXならコマンドの後ろに&記号をつけるだけでこうなります。コマンドを入力すると，新しいタスクとして打ち出すプログラムが登録されるだけで，すぐにコマンド待ちに復帰します。次のコマンドを入れればちゃんと動くところからも，プリンタへの打ち出しコマンドを実行している姿はまったく見えませんが，ちゃんとプリンタはジージーと鳴りながら動き続けているのです。

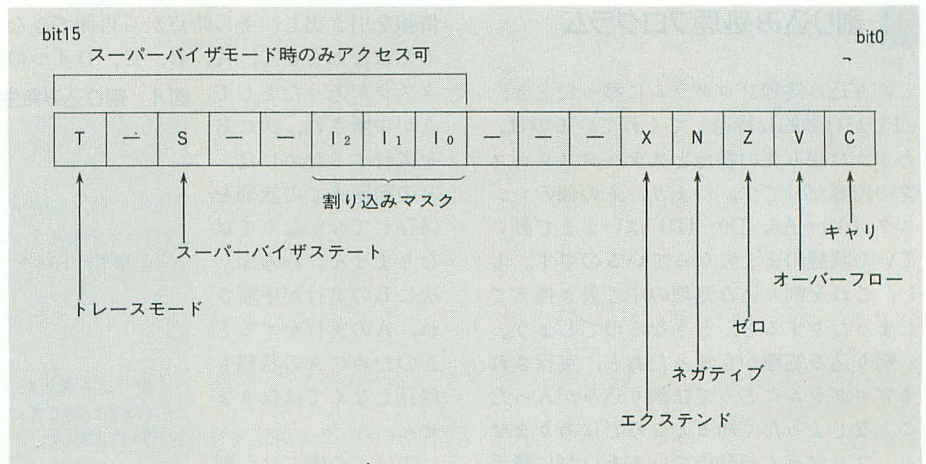
UNIXのコマンドなどを見ていると，さらにマルチタスク機能を便利に使っている例をいくらも見ることができます。たとえば，一般のパソコン用のCコンパイラは複数のパスに分かれていますが，これらをひとつずつ動かすよりないため，必ずディスク上に大きな作業用ファイルをいくつも作っては次のプログラムを読み込み，処理しては次のファイルを作り，前のを削除する，といったことを行います。このため，ディスクアクセスが非常に多くなることになります。

UNIXなどでは，Cコンパイラを起動すると，裏では必要なタスク（UNIXではプロセスと呼んでいますが，ここではタスクで統一します）を一斉に起動し，互いのプログラムのあいだをパイプというタスク間の通信路で結んでしまいます。1つひとつのタスクは自分の入力として入ってきたデータを処理し，自分とパイプでつながれたタスクに渡していくという方法で，コンパイル作業が行われているのです。パイプで結んでおけば，複数のCコンパイラが同時に動かされたとしても（なにせ，マルチタスクですから）中間ファイルの名前がぶつかってわけのわからない動きをすることもありませんから，メモリをやたらと食うという以外はうまい方法といえるでしょう。

68000の割り込み機能

それでは，本命の68000の割り込みを見ていきましょう。68000の場合，これまで触れてきたような割り込み処理と同様の動きをするものとして，アドレスエラーやバスエラー（変なことをすると画面の真ん中にドドーンと表示されることがありますね），ソフトウェアのTRAP命令（X68000のIOCSコールはこれを使っています）なども含めた呼び方として「例外（エクセプション）」と呼んでいますが，ここではハードウェアによる割り込みを中心に考えますので，「割り込み」という用語で統一していくことにします。

68000の割り込み動作はZ80や8086などよりもかなり凝った作りになっています。アーキテクチャ上，大きく違うのは，68000にはスーパーバイザモードと，ユーザーモードの2つの動作モードがあることと，割り込みに7段階のプライオリティ（優先順位）を設けていることです。

スーパーバイザモードではあらゆる命令が実行できますが，ユーザーモードではシステムの動作に直接影響を与えるような命令の使用が禁止されます。さらにX68000では，ユーザーモードではVRAMを含め，各種の周辺コントローラやOSの領域へのアクセスが行えないようになっています。どちらの場合にも禁止されていることを行おうとした時点で自動的に割り込み動作が行われ，処理プログラムが実行されます。

また，プライオリティ管理は，ステータスレジスタの上位バイトにある3ビットのマスクによって制御されます（図3参照）。Z80や8086では割り込みは禁止と許可の2段階しかありませんでしたが，68000では7段階まで拡張されているため，マスクビットも3ビット必要になっています。このビットは割り込みが入ったときにそのプライオリティと同じ値にセットされ，再び割り込み要求があっても，この3ビットで決められた以上のプライオリティでなければ受け付けられません。

逆にいえば割り込み処理中であっても，よりプライオリティの高い割り込みがあれば，それを受け付けるようになっているのです。これによって，ある程度ゆっくりと処理を行ってよい割り込みが処理されているために，緊急度の高い割り込み処理ができなくなるようなことを避けられることになります。

割り込みが入ってからの68000の動きはZ80のモード2割り込みや8086の割り込み動作とよく似ており，おおむね次のようになります。

まず，68000はただちにスーパーバイザモードに移行し，スタックポインタ（A7レジスタ）がスーパーバイザ用のもの（システムスタックポインタとも呼ばれます）とすりかわります。Z80や8086ではスタックポインタはひとつだけでしたが，68000ではモ

図3　68000のステータスレジスタ

ード別に別々のスタックポインタを持っているのです。

次に割り込み処理が終わったときに戻るべきアドレスと，現在のステータスレジスタの値をシステムスタックに待避します(図4参照）割り込み処理が終わったときにRTE命令を実行すると，この番地に戻っていくため,割り込まれた側のプログラムはなにごともなかったかのように処理が継続できるわけです。

待避が終われば，次にCPUは割り込みベクタの読み出しを行います。ベクタ番号は，今回の割り込みの要因がなにであるかを識別するためのもので，値としては0からFFHのあいだになります。さらに，読み取ったベクタをもとにベクタテーブルを参照し，そこに書いてある割り込み処理プログラムのアドレスを得ます。

ベクタテーブルはメモリの0番地から書いてあり，アドレスはひとつあたり4バイトですから，(読み取ったベクタ)×4番地の内容が割り込み処理プログラムのアドレスということになります。このあたりは8086などとまったく同じです。

割り込み処理アドレスがわかれば，そこから順次プログラムを実行していき，最後にRTE(ReTurn from Exception）命令を実行すると，システムスタックから帰り先の番地と割り込みが起きた時点でのステータスレジスタの値を取り出してそこに戻るわけです。

68000で発生する割り込みとしてはこのほかにもTRAP命令など，ソフトウェアによるもの（DOSコールやIOCSコールを行うのに使われています)，外部から異常が伝えられるバスエラー，奇数番地からワードで読み出そうとしたような場合に発生するアドレスエラーなど，数々の割り込み要因があります。

割り込み処理プログラム

割り込み処理プログラムに移ったとき，CPUが自動的に待避してくれているのは，わずかに帰り先の番地とステータスレジスタの内容だけです。つまり，その他のレジスタ（A0～A6, D0～D7）はいままで動いていた状態のままになっているのです。もし，これを割り込み処理の中で書き換えてしまったりすると，どうなるのでしょう。

割り込み処理から帰ったあと，実行されるプログラムにとっては割り込みが入ったことなどまったく知るところではありません。プログラムが動いているあいだに勝手にレジスタの内容が変わってしまい，そのまま実行を続けてしまうことになるのですから問題です。運が悪ければ暴走にもつながりかねません。

むろん，割り込み処理プログラムの中でまったくレジスタを使わないというのであれば，問題はないのですが，それではほとんど満足な処理は行えないでしょう。したがって，割り込み処理プログラムの中でまずやらなくてはならないことは自分の中で使うレジスタをセーブすることです。

いちばん簡単なのは現在のスタック，すなわちシステムスタックに全レジスタの内容をセーブし，RTEを実行する直前に元に戻してやる方法です。つまり，

```
int_entry:
        movem.l d0 -d7/a0-a6,-(sp)
            :
            :
        movem.l (sp)+,d0-d7/a0-a6
        rte
```

とやるわけです。これで戻ったときには全レジスタの内容が変化せずにいることになりますから，無事になにごともなかったようなフリができるわけです。雑誌に掲載されるプログラムでも，バックグラウンドで音楽を鳴らしたり，絵を動かすといった，割り込みを横取りして動くようなタイプのものはこのような方法をとっています。

しかし，リストを見ていただければわかるとおり，今回の割り込み処理ではこの方法はとっていません。時分割処理を行うにはこの方法ではうまくいかないのです。次にこの点について検討してみることにしましょう。

時分割処理と割り込み

時分割処理を行うためには，2つのタスクの状態を保存し，任意の側の前回までの情報を引き出し，その時点から再開できなくてはなりません。A，B，C，D4つのタスクがあったとして，Aが中断され，次にBを実行するためには，Bの前回までの状態を保存しておかなくてはなりません。むろん，次にBの実行が中断され，Aの実行をするときのためにAの状態も保存しなくてはなりません。

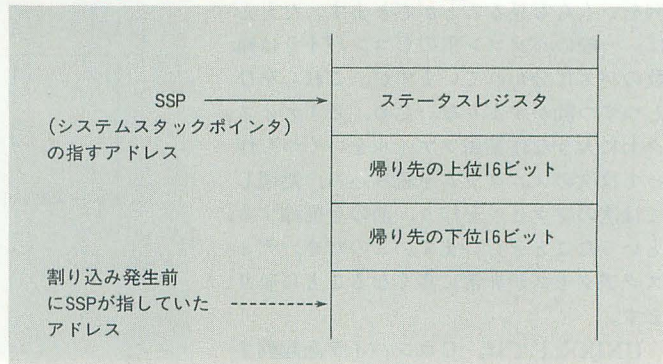
先ほどの例では，割り込みが入った時点のレジスタの内容をすべてシステムスタックに積み上げていました。この方法を使ったとしましょう。いままで動いていたタスクをAとして，次にBに移るためにはどうしたらよいのでしょうか。また，Bになんらかの方法で移ったとして，いましがたシステムスタックに積み上げたタスクAのレジスタの内容はどうすればよいのでしょう。

Bのレジスタは，必ずいまのAのレジスタ情報のさらに下にあるという方法を考えれば，考えられなくもありませんが，いかんせん融通がきかない方法であることは間違いありませんし，なにより貴重なシステムスタックが，各タスクのレジスタの待避エリアに使われるのはなんとも面白くありません。

この解決策として考えられるのは，各タスクに1対1で対応するテーブルを作ることです。選び出したタスクの実行を再開させるときは，そのテーブルから情報を引き出し，各レジスタの内容をすべて前回の時点での値に復帰させたあと，RTE命令で戻ればよいわけです。このようなテーブルをTCB(タスクコントロールブロック）とか，プロセステーブル（UNIXの場合）と呼んだりしています。

レジスタの情報の待避する場所としてよく使われるのはTCBや各タスクのスタックです。スタックに保存した場合は，最後のスタックの位置だけをTCBに保存しておけば，残りのレジスタの値はそこから順に引っ張り出してくることができることになるわけです。どちらを使うかは設計者しだいといったところでしょう。

TCBにタスクのすべての情報を残すという点ではレジスタの内容を全部保存するほうがよいでしょうが，すでに保存領域が指し示されているのだからスタックを使うほうが簡単であるし，TCBのサイズが圧縮できるではないかというのも確かに理屈ではあります。今回は後者の方法を採用してみ

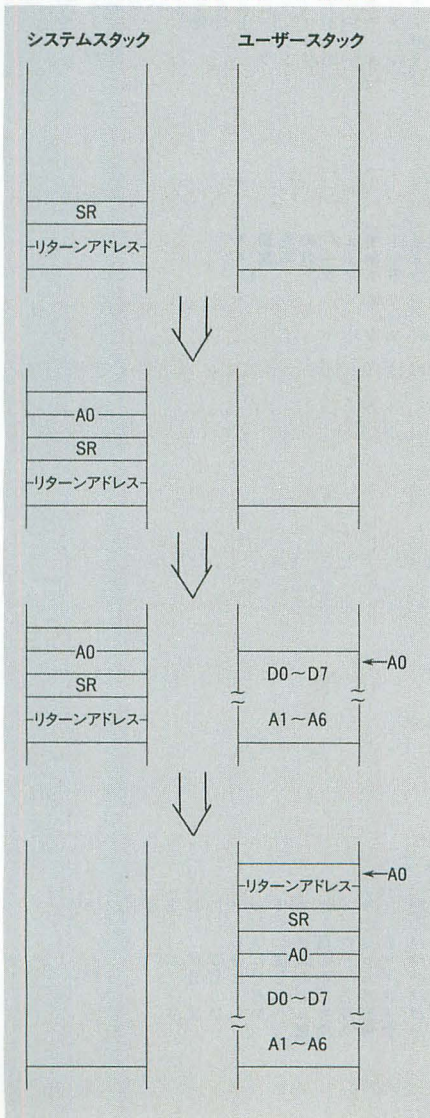
図4　割り込み発生時のシステムスタックの状態

ました。スタックのオーバーフローなどを考えると若干前者に分があるかなという感じもしないではないのですが……。

まず，割り込みが入った時点で一時的にワークとして使うレジスタをシステムスタックにセーブします。

ここではA0がそれにあたります。次にA0にユーザースタックの内容をコピーし，ユーザースタック領域をA0で指し示します。ここで，残るレジスタD0～D7，A1～A6をmovem命令を使って一気に積み上げていきます。さらに，先ほどシステムスタックに積み上げておいたA0の値と，割り込み発生時にCPUによって自動的に積まれていたステータスレジスタと帰り先の番地を積み上げれば，全レジスタのセーブが完了します。このときのスタックの動きを図5に書いておいたので，参考にしてください。

ここで，ステータスレジスタのチェックが入っているのは，X68000のIOCSコール

図5　割り込み発生後のスタックの動き

リスト1

```
==================  TSS.S  ==================
  1: *
  2: * 時分割処理サンプルプログラム
  3: *
  4: *
  5: * 1990-05-24 Written by M.kuwano
  6: *              No rights reserved.
  7: *
  8:   .globl        _task_1,_task_2,_task_3
  9:   .xref         _tss0,_ts_init
 10:   .include      iocscall.mac
 11:   .include      doscall.mac
 12: FLAG_SUPER      equ     5                   * スーパーバイザか否かを示すフラグ
 13:
 14:         .text
 15: sema:           dc.b            0
 16:
 17:         .even
 18:
 19: *
 20: *  割り込み処理
 21: *
 22: tss:    move.l          a0,-(sp)            * スーパバイザスタックを拝借
 23:         move.l          USP,a0              * ユーザスタックの取り出し
 24:         movem.l         d0-d7/a1-a6,-(a0)   * a0以外をユーザスタックに待避
 25:         move.l          (sp)+,-(a0)         * a0も待避
 26:         move.w          (sp)+,-(a0)         * フラグ
 27:         move.l          (sp)+,-(a0)         * リターン・アドレス
 28:         bsr             chk_flag            * フラグをチェック
 29:         bne             tss_pass            * 既にスーパーバイザだったらパス
 30:         move.l          a0,-(sp)            * 引き数としてA0の値を渡す
 31:         bsr             _tss0               * スケジューリング
 32:         addq.l          #4,sp
 33:         move.l          d0,a0               * 値はD0に入ってくる
 34:   tss_pass:
 35:         move.l          (a0)+,-(sp)         * 帰り先を積んで
 36:         move.w          (a0)+,-(sp)         * フラグも積んで
 37:         move.l          (a0)+,-(sp)         * a0を一時待避
 38:         movem.l         (a0)+,d0-d7/a1-a6   * レジスタ復帰
 39:         move.l          a0,USP              * ユーザ・スタックのすりかえ
 40:         move.l          (sp)+,a0            * a0も復帰
 41:         rte
 42:
 43:
 44: *
 45: *  タスク0～3
 46: *
 47: _task_0: bsr            _ts_init
 48:         lea.l           tss,a1
 49:         move.l          #$7,d1
 50:         IOCS            _VDISPST            * 垂直同期の割り込みを使う
 51: loop:
 52:         move.w          #$31,d1
 53:         bsr             get_sema
 54:         IOCS            _B_PUTC
 55:         bsr             put_sema
 56:         bsr             wait
 57:         bra             loop
 58:
 59: _task_1: move.w         #$32,d1
 60:         bsr             get_sema
 61:         IOCS            _B_PUTC
 62:         bsr             put_sema
 63:         bsr             wait
 64:         bra             _task_1
 65:
 66: _task_2: move.w         #$33,d1
 67:         bsr             get_sema
 68:         IOCS            _B_PUTC
 69:         bsr             put_sema
 70:         bsr             wait
 71:         bra             _task_2
 72:
 73: _task_3: move.w         #$34,d1
 74:         bsr             get_sema
 75:         IOCS            _B_PUTC
 76:         bsr             put_sema
 77:         bsr             wait
 78:         bra             _task_3
 79:
 80: *
 81: *  時間待ちループ
 82: *
 83: wait:
 84:         move.w          #$ffff,d0
 85:   dummy:
 86:         dbra            d0,dummy
 87:         rts
 88:
 89: *
 90: *  既にスーパーバイザモードになっていたかのチェック
 91: *
 92: chk_flag: btst.b        #FLAG_SUPER,4(A0)
 93:         rts
 94:
 95: *
 96: *  セマフォ獲得  取れるまで無限ループ
 97: *
 98: get_sema: bset.b        #1,sema
 99:         bne             get_sema
100:         rts
101:
102: *
103: *  セマフォ解放
104: *
105: put_sema: bclr.b        #1,sema
106:         rts
107:
108:         .end            _task_0
```

やDOSコールの中でも割り込みが入ってしまうためです。すべての割り込みを自分のプログラムで管理しているのなら，ここでは割り込みがあったというフラグだけを立てて，もともと実行していたIOCSコールなどの処理を行い，最後にいまの割り込みの処理をしておくといったテクニックも使えるのですが，横取りした割り込み以外はすべてHumanでセットされたままで使う関係上しかたがないので，すでにスーパーバイザモードにいるようであれば，なにもせずに元に戻るようにしています。

レジスタの待避が終わったらA0の値をもってTSS0というサブルーチンを呼び出します。TSS0はわかりやすいようにCで書いてみました。もちろんアセンブラで書くこともそれほど難しくはありませんが，見やすさという点ではCのほうがよいでしょう。TSS0では動かしていたタスクの番号からTCBのアドレスを計算し，その中に渡されたA0の位置を保存します。

あれ？ USP（ユーザースタックポインタ）は？ と思われた方も多いでしょう。アセンブラのほうでTSS0を呼び出したあとからRTE命令までの部分を読んでみてください。最後に積み上げた場所から順次データを引き抜いていくって，最後に残ったA0の値，これは元のスタックの位置にほかなりません。ここで，A0をUSPに移し，さらにA0自身の値も元に戻せばめでたしめでたしとなるわけです。このときのスタックの動きは，ちょうど割り込みが入ったときと逆になります。

次に各タスクの選択のやり方を見てみることにしましょう。0番のタスクは起動したときから動いていたプログラム，その他のものは順番がくるのを待って1列に並んでいます(図6（a）参照)。この先頭を指しているのがrdy_q_head，最後を指しているのがrdy_q_tailです。

現在実行しているタスクはc_taskで指し示されるTCBのものです。当然，c_taskの初期値はTCB[0]を指すようになっています。ここで割り込みが入ると，TCBのつなぎ換えが行われます。いままで実行されていたタスク（c_taskの指しているTCBのタスク）のTCBはrdy_q_tailが示すTCBの後ろにつながれ，代わりにこれまでrdy_q_headが指していた，キューの先頭のTCBをc_taskが指すようになります。rdy_q_headは，いままでキューの2番目にいたTCBを指すように変更されます(図6（b）参照)。

アセンブラプログラム中のget_sema，put_semaの2つは，シングルタスクでのプログラムしか見慣れない方にはちょっと珍しいコーディングかもしれません。これは，セマフォと呼ばれるプログラミング手法のひとつで適当なプログラム領域の先頭と末尾でget_semaとput_semaを実行することで，これに挟まれた区間は一度には必ずひとつのプログラムしか実行しないようにするしかけです。

get_semaで使われているbset命令は，単に指定されたビット位置を1にするだけでなく，そのビット位置の前回値が0であったか否かによってZフラグが変化します。semaの領域は初期値として0が入っています。いま，タスクAがここに飛んできたとしましょう。bset命令によって，semaが1，Zフラグはもともと0でしたから1になります。Zフラグが立っていますから，そのまま下に向かい，IOCSコールが行えます。

リスト2

```
================== TSC.C ==================

 1: /*
 2:  *          時分割処理サポートルーチン
 3:  *          ＴＣＢ管理＆初期化
 4:  *
 5:  *          1990-05-24  Written by M.kuwano
 6:  *          No rights reserved.
 7:  */
 8:
 9: #define   READY          0
10: #define   EOQ            -1       /* キューの最後を示す  */
11: #define   NR_TASKS       4        /* タスクの数          */
12: #define   TASK_STK_SIZE  0x100    /* タスクのスタック    */
13:
14: extern    void    task_1(),task_2(),task_3();
15:
16: struct    TCB     {
17:   unsigned char   state;          /* タスクステート（今回は使っていない） */
18:   unsigned char   sstate;         /* サブ・ステート（    上に同じ      ） */
19:   unsigned int    *usp;           /* ユーザー・スタック                 */
20:   struct  TCB     *tcb_link;      /* レディキューのリンク               */
21: } tcb[NR_TASKS];
22:
23: struct    STACK {
24:   unsigned int    task_sp[TASK_STK_SIZE];
25:   void            *task_pc;
26:   unsigned short  task_sr;
27:   unsigned int    task_regs[16];
28: } t_stack[NR_TASKS];
29:
30: struct    TCB     *rdy_q_head;    /* レディ・キューの先頭 */
31: struct    TCB     *rdy_q_tail;    /* レディ・キューの末尾 */
32: struct    TCB     *c_task;        /* カレントタスク       */
33:
34: void ts_init()
35: {
36:   unsigned int    i;
37:   struct  TCB     *t;
38:   struct  STACK   *s;
39:   t_stack[1].task_pc = task_1;
40:   t_stack[2].task_pc = task_2;
41:   t_stack[3].task_pc = task_3;
42:   c_task = tcb;
43:   c_task ->state = READY;
44:   c_task ->sstate = 0;
45:   c_task ->tcb_link = EOQ;
46:   t = rdy_q_head = &tcb[1];
47:   s = &t_stack[1];
48:   for (i=1; i<NR_TASKS-1; i++, t++, s++) {
49:           t ->state    = READY;
50:           t ->sstate   = 0;
51:           t ->usp      = &s->task_pc;
52:           t ->tcb_link = &tcb[i+1];
53:           s ->task_sr  = 0;
54:   }
55:   t ->state    = READY;
56:   t ->sstate   = 0;
57:   t ->usp      = &t_stack[i].task_pc;
58:   t ->tcb_link = EOQ;
59:   s ->task_sr  = 0;
60:   rdy_q_tail = t;
61: }
62:
63: void *tss0(stack)
64:   unsigned int    *stack;
65: {
66:   struct  TCB     *p;
67:   c_task ->usp = stack;                       /* スタックをＴＣＢにしまう   */
68:   if (rdy_q_head != EOQ) {
69:           c_task ->tcb_link = EOQ;            /* 自分が最後になる           */
70:           rdy_q_tail ->tcb_link = c_task;     /* キューの最後につなげる     */
71:           rdy_q_tail = c_task;                /* キューのおしりは自分       */
72:           c_task = rdy_q_head;                /* 次の方、どーぞ             */
73:           rdy_q_head = c_task ->tcb_link;     /* 先頭をキューからはずす     */
74:           c_task ->tcb_link = EOQ;            /* 気分の問題                 */
75:   }
76:   return(c_task->usp);
77: }
```

これより少し遅れて次のタスクBがここにきたとします。同じようにbset命令を使うと，今度はすでに1が立っていますからZフラグが立ちません。したがって，ここでbsetをやり続けることになるわけです。

さて，タスクAが処理を終わり，put_semaを実行したとします。ここでsemaの値は0に戻ります。そして，次にタスクBがget_semaをやると今度はZフラグが立つので，無事にループが止まり，下に抜けてくるわけです。

このようなことが行えるのは，割り込みというのが（たとえNMIであろうと），1命令の処理の途中では入ってこられないためなのです。bset命令はメモリからのデータの読み込みのあと，再び書き込むという，2段階で実行されますが，この隙間には絶対割り込みは入らないのです。もし，ここをCなどで書くと，おそらくmove命令が複数個並ぶことになるでしょう。こうなると，セマフォの役をはたさなくなってしまいます。

もし，タスクAがsemaの値を持ってきたところでタスクBに切り替わり，ここでもタスクBがsemaを読み出したとします。タスクA，Bの両方ともsemaの値としては0を読み出すことになるため，セマフォが取れたつもりになってしまいます。

8086なら，ここで一時的に割り込みを禁止するなどという，野蛮な手法も使えるのですが，68000では割り込みフラグの操作は特権命令ですから，スーパーバイザモードでなくては使えません。各タスクはユーザーモードで動いていますから，もしこんなことをすればただちに特権命令違反になってしまいます。

このようなクリティカルな部分が記述できるのはやはりアセンブラならではといえるでしょう。

動かしてみよう

サンプルのソースを入力してコンパイル，アセンブル，リンクしてください（cc tss.s tsc.c とでもすればよいでしょう）。エラーメッセージが出なくても，打ち間違えや行の抜けなどがあるとあっさり暴走してくれますので，外付けならば実行前にハードディスクの電源はOFF（念のため，ブレイクキーを押してからにしましょう），フロッピーディスクも抜いておいたほうが賢明でしょう。

うまくいけば，1から4の数字（タスクの番号と一致しています）がゾロゾロゾロ……と表示されるはずです。タスク0からは縁もゆかりもないまったく別のところにあるプログラムがちゃんと同時に（正確には順番にですが）動いているのがわかることでしょう。ブレイクキーセンスなどはやっていませんので，やめたくなったら，リセットしてください。

うまく動いたようでしたら，いろいろと改造してみるとよいでしょう。X68000の場合，グラフィックなどもIOCSでサポートされていますから，マルチタスクでグラフィック画面にいろいろ表示させるというのも簡単なことです。

おわりに

アセンブラでのプログラミングのサンプルというと文字表示やら簡単な計算やらが多いのですが，今回はアセンブラならではということを考え，68000の割り込み処理をつつき，簡単な時分割処理を行ってみました。

単なる時分割ですから，タスク間の通信も，ファイルやメモリの管理もありません。タスクの状態遷移という考えすらないごく原始的なものではありますが，実のところ，巨大なマルチタスクOSもそのカーネル部分まで下がってみると，最後はこの程度のレベルに落ちつくのです。

要するにどんな巨大なOSであっても，最下層のところでは複数のプログラムの状態を記録しておいて，割り込み（エクセプションといったほうが正しいかもしれませんが）が入ったときにそれらの中に現在実行していたタスクの状態を記録し，次に動作させるタスクの記録を引っ張り出してきてそれの実行にとりかかるというだけのことなのです。

大胆不敵にいってしまうと，この段階がわかってしまえばマルチタスクの気分が味わえる程度のOS（モドキ）ならそれほど苦労しなくても作れてしまうのではないでしょうか。

いっそ，夢はでっかくUNIXライクなパーソナルOSを作ろう会！ ……誰かやる人いませんか？

図6　タスク切り替えの考え方

S-OS&REDAを忘れていませんか

Z80マシンとして愛用されてきたX1，MZ。もはや語り尽くされてきたマシンとはいえ，アセンブラはあきらめちゃったという人も多いはずです。いま一度，眠っているS-OSのディスクを引っ張り出してマシン語に挑戦してみてはいかがでしょうか。

Yamada Junji
山田 純二

X1やMZユーザーの皆さん，お元気ですか。大好きなマーベルランドのCDを買ったら，予約もしていないのにポスターをもらってウキウキしている山田純二です。

さて，6月号発表のアンケート結果を見ると，X1，MZユーザーの40%もの人がアセンブラを使っているそうです。自力でマシンをコントロールできれば8ビットマシンでもいろいろと楽しめますからね。とはいえ，マシン語をやろうと思い立ってS-OS"SWORD"用のアセンブラREDAやZEDAなどを打ち込んだけど，結局投げ出してしまったという人も多いでしょう。今回は，これらの使い方について，もう一度だけ基礎の基礎から説明していくつもりです。いったんあきらめてしまった人も，実はこれからと思っている人も，挑戦してみてください。もうアセンブラは，マスターしたという人も，昔を懐かしみながら読んでみてもらえると，僕はとてもうれしいです。

使用システムはS-OS"SWORD"，アセンブラは，REDAを使います。ほかにも，OHM-Z80などや，今月号のWZDなどがありますが，多機能すぎて初心者向きではない（と僕は思った）ので，今回は，REDAを使って話を進めていこうと思います。

図1 S-OSシステム関係図

リスト1

```
 1
 2 ;SAMPLE1
 3 #DRDSB  EQU     $2000
 4
 5         ORG     $9000
 6 HAJIME
 7         LD      A,01
 8         LD      HL,INPUT
 9         LD      DE,(NO.)
10         CALL    #DRDSB
11         RET
12 INPUT   DS      256
13 NO.     DW      25
```

超入門編

●STEP1 モニタでなにをするのか

まず，"SWORD"を起動すると，プロンプト"#"が表示されます。これがS-OSのモニタと呼ばれるコマンドモードで，基本的なファイル操作を行ったり，プログラムの実行をさせます。S-OSのモニタに対するコマンドはすべて，この"#"の後ろに続けて書かなければなりません[1)]。

●STEP2 モニタサブルーチンとは

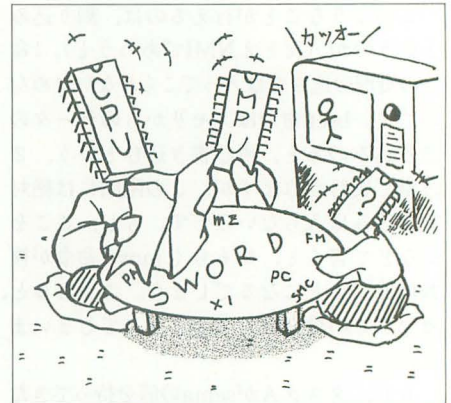
多機種にわたる"SWORD"上で，アプリケーションを作成するには，機種の違いをなくすため用意された，共通ルーチンを使う必要があります。システムの上下関係は図1のようになっています。これを見れば，なぜ，わざわざモニタサブルーチンを使わなければならないのかおわかりでしょう。

サブルーチンの具体的な使用例として，リスト1を見てください。ここでは，ディスクからの連続セクタリードを行うサブルーチンをコールしています。まず，サブルーチンコールに必要なパラメータとして，

A……読み込むレコード数
DE……読み込む先頭レコードナンバー
HL……読み込み先のアドレス

をそれぞれ指定されたレジスタへ渡してから，2000Hのアドレスをコールします。サブルーチンコールは，すべてこのようにして使います。それぞれのサブルーチンによって破壊されるレジスタが違うので，必要に応じてレジスタを保存しましょう。

●STEP3 いよいよアセンブラ

それでは，実際にリスト1をアセンブルしてみましょう。初めに，モニタ上からREDAを読み込み，J3000で起動させます。最初は，アセンブラモードで立ち上がりますから，"E"コマンドでエディタのコマンドモードに移ります。ここでは，テキストの初期化，テキストファイルのセーブ/ロードなどが行えます。新しくテキストを入力する場合は，"N"コマンドによって，テキストエリアを初期化しておいたほうが安全です。

テキストの編集は"E"コマンドで，エディタモードへ移ってから，リスト1を入力しましょう。入力が終わったならば，いったんデバイス（ディスク）にセーブしてから，シフト＋ブレイクでアセンブラモードへ戻り，

：A/

とすれば，アセンブルできます。"A"コマンドの後ろのスラッシュは，アセンブル時に，画面へアセンブルリストを出力させるものです。慣れないうちは，リストを表示させながらアセンブルしたほうがいいでしょう。なぜかって？ それは，オブジェクトとソースリストの関係がよくわかるからです。

アセンブルエラーが出ると，自動的にエディタモードへ移ってエラーの出た行を表示してくれるので，元のリストとよく見比べて修正後，再度アセンブル。無事アセンブルが終了すると，使われたラベル数と，スタートアドレス，エンドアドレス，エントリアドレス，オフセットアドレス，を表示して完了です。

実行は，アセンブラモードか，"SWORD"

1) モニタコマンドについては，"SWORD"掲載号の記事を見れば一覧表が載っています（たとえば，先月号の125ページ）。

のモニタ上から，J9000です。

●STEP4　ソースリストの構成と疑似命令

では，ひととおりアセンブラの扱い方がわかったところで，ソースリストがどのように記述されているか説明していきましょう。これがわかっていなければ，ソースリストは，読めませんからね。

ステートメントは，大きく分けて3つの部分に分類でき，左から，ラベル，オペコード，オペランドから構成されています。

まず，ラベルですが，これは必ず文頭に書かなければなりません。ひとつでも，空白があるとアセンブラはそれをオペコードと解釈してしまいます。オペコードには，コンピュータに対する命令を書きます。オペランドとは，オペコードで書いた命令を，詳しく規定するものです。オペコードとオペランドの間は，ひとつ以上の空白が必要です。

これら3つの区切りは厳密に規定されてはいませんが，リストの保守性を高めるためにも，自分なりに綺麗に区切っておくといいでしょう。もうひとつ，“；”の書かれたあとは，行末まで，コメント行として解釈されます。

さて，ソースリストを読むためにはもうひとつ，疑似命令というものも知っておかなければなりません。疑似命令というのはアセンブラに対する命令で，プログラムの実行には関係はありませんが，プログラムを作るときには必要不可欠なので，覚えておきましょう。

・ORG命令

アセンブラが生成する，オブジェクトプログラムの，先頭アドレスを指示する命令です。

・ラベル

BASICで使うラベルと同じようなものですが，アセンブラでは，単に目印となるだけではなく，ラベルにプログラムのアドレスが定義される，という意味があります。

・EQU命令

ラベルにオペコードの値を定義する命令。値は必ず2バイトで，指定しなければなりません。

・データ定義命令

DEFB……1バイトのデータを格納
DEFW……2バイトのデータを格納
DEFS……オペランドで指定した数だけメモリの領域を確保
DEFM……オペランドに指定した，文字列をメモリに格納

以上の4つがもっとも重要な疑似命令です。

さあ,実践だ！

それでは，モニタサブルーチンの使用法の説明もかねて，ちょっと役に立つ可愛いサブルーチンを作っていきます。“SWORD”で扱えるファイルには，テキストファイルとオブジェクトファイルの2種類ありますが，モニタ上では，テキストファイルが扱えません[2)]。ここはやっぱり，扱えないのなら，自分で扱えるようにしてしまえの精神で，外部コマンドとしてのTYPEコマンドを実際に作ってみましょう。

●メインルーチンの作成

まず，リスト2（サブルーチン名BEGIN）の説明です。これは，起動したときのメッセージ表示と，のぞきたいファイル名を入力するルーチンです。メッセージの表示方法としては，ここで使っている#MPRINTのほかに#MSG，#MSXなどがあります。

#MPRINTというのは，コールした次のアドレスから00_Hがあるまで，ASCIIコードとみなして文字列出力します。#MSGはDEレジスタに示すアドレスから，$0D_H$があるまでASCIIコードとみなして文字列出力。#MSXは，#MSGと大体同じで文字列のエンドコードが00_Hです。

3つのサブルーチンに共通するのは，文字列の表示後に改行しないということです。改行させたい場合は，表示したあとに#LTNLをコールするか，文字列中に改行コードの$0D_H$を入れておく必要があります。

で，お次はファイル名の入力。これは，キーボードからの1行入力ルーチンを使います。DEレジスタにキー入力バッファの先頭アドレスを入れてからコール。バッファは，最低でも80バイト必要ですので，自分でバッファを用意するか，各機種のモニタに用意されているバッファを使いましょう。

各機種用のバッファアドレスは，S-OSのワークエリアの#KBFADに格納されています。注意しなければならないのは，画面の左端からバッファに取り込まれることです。メッセージを表示して文字列入力したときには，入力後に，メッセージ分の文字列をスキップさせてやらなくてはなりません。これは，出力されてくるDEレジスタはバッファの先頭アドレスを示しているからです。リスト2の場合，メッセージは10文字使っているので，DEレジスタに10文字分足してから，ファイル入力ルーチンへ渡しているのがわかるでしょう。

●ファイルの入力

S-OSでファイルの入力をする場合は，以下に示す手順で行います。ファイル入力ルーチンはリスト3，サブルーチン名はFILE READです。これを見ながら読んでいってください[3)]。

1)　ファイル名の入っている先頭アドレスをDEレジスタにセット，Aレジスタにファイルのアトリビュートをセットしてから，#FILEをコール。
2)　次に，ファイルの情報を取り出すために，#ROPENをコール。
3)　#ROPENで取り出したファイル情報に従って，デバイス上のファイルを読み込む。これは#RDDをコールします。ちゃんと#ROPENをコールしてファイルをオープンしないとエラーとなってしまうので注意してください。
4)　おわり。

1)のファイルのアトリビュートとは，読み込むファイルの属性を指定するもので，オブジェクトファイルの場合は1，テキストファイルの場合は4と指定します。この場合はテキストファイルを扱うので，Aレジスタに4をセットして#FILEをコールしています。そして，#ROPENでファイル情報の取り出しを行い，その後，強制的にファイルのロード先を変更しています。

ロード先の変更は#ROPEN後，#DTADRを変更すれば，ファイルは新しく指定されたアドレスにロードされます。E-MATEなどはファイルのロード先をオフセットでセーブしているため，そのままだと0000_Hにテキストをロードしてしまうので，#ROPEN後に変更してやらねばなりません。エラーがあるとキャリフラグをセットして帰ってくるので，そのときは#ERRORにジャンプさせるようにしておきましょう。

●テキストの表示ルーチン

ファイル入力も終わり，次はリスト4（サブルーチン名TYPEMAIN）の読み込んだテキスト表示の説明です。テキストの構造は，文字列，改行コード$0D_H$，の順番に1行ごとに並んでいて，エンドコードは00_Hとなっています。これを知っていれば，1行ご

2)　テキストファイルはアプリケーション用の文字列ファイル，オブジェクトファイルは実行可能なマシン語プログラム，またはマシン語データとなっています。

3)　このTYPEルーチンは，一度メモリに読み込んでから，ドバーっと表示するというあまり賢くない構造です。本来なら1クラスタごとに読み込んで，そのつど表示していくのが正しい姿ですが，根性のある人は，自力で挑戦してみてください。

とに改行して，エンドコードを調べながら，テキストを文字列表示する必要はなく，ドバーっと文字表示を行っていけば，1行ごとに自動的に改行してくれるのがわかるでしょう。

1文字表示は，AレジスタにASCIIコードをセットして#PRINTルーチンをコールすれば，1文字ポコンと表示されます。表示をやりっぱなしだと，ちょっと困ったちゃんなので，#PAUSEを使ってテキスト表示の一時停止機能をつけてあります。#PAUSEの機能は，スペースキーが押されたら何かキーを押すまでリターンせず，シフト＋ブレイクが押されると，コールされた次の2バイトに示されるアドレスにジャンプします。この場合はブレイクを押されたら表示を終了させたいので，TM3をセットしてあります。

以上で，リストの解説はおしまい。それぞれのリストをBEGIN，READ，PRINTのファイル名でセーブし，アセンブルするときは，REDAのアセンブルモードから

:ABEGIN:READ:PRINT

と打ち込むとアセンブルできます。実行は，リスト1のサンプルと同じ方法で行ってください。

●オフセットをつけてみよう

リストをそのまま打ち込んで，アセンブルすると9000Hにオブジェクトが生成されますが，メモリのど真ん中にプログラムを置くのはいまいち気持ち悪いので，今度は，3000Hからプログラムを生成させるようにしてみましょう。

これは，素直にORG命令のところを3000Hに変更しただけでいいように思いますが，3000Hにはアセンブラが，でーんと居座っているので，ORG命令で，3000Hにしてしまうとアセンブラが破壊されてしまいます。これを避けるには，3000Hから動作するプログラムを，一時的にシステムと重ならないような別の場所にアセンブルしてやらなければなりません。そのための疑似命令として，OFFSET命令があります。使用法は，

OFFSET $6000
ORG $3000

のようにして使います。ORG命令で指定したアドレス＋OFFSET命令で指定したアドレスにオブジェクトを生成することができるのです。この場合だと6000H＋3000H＝9000Hのアドレスに，3000Hで動作するプログラムが出来上がります。

さて，このプログラムを実行するためには，9000Hから生成されたオブジェクトを，ORG命令で指定したアドレスにロードしなおす必要がありますが，REDAで分割アセンブルする場合は，最後のファイルセーブで自動的にこれらの処理をやってくれるので，あまり気にしなくていいかもしれません。というところで，サンプルのプログラムの変更点は，リスト2のBEGINの8行目に，

OFFSET $6000

の1行をつけ加えて，9行目のORG命令のアドレスを3000Hに変更してからアセンブルしてください。アセンブルが終わると，セーブするかどうか聞いてきますので，“Y”と入力してセーブしておきましょう。この場合，ファイル名は，

BEGIN.OBJ

で強制的にセーブされてしまいます。実行は，いったんBEGIN.OBJをS-OSのモニタからロードしなおして，

#J3000

で実行してください。

さて，実例をまじえてアセンブラとS-OS“SWORD”の使い方を説明してきました。いかがだったでしょうか。まだ8ビットで頑張ろうという人はぜひともマシン語を覚えて愛機を活用してください。

自分でやっちゃえ！

ノーマルのまま“SWORD”のモニタを使っていると，慣れるにしたがって，いろいろと不満が出てくることでしょう。そんなときには，このように外部コマンドとして，どんどん自分でシステムを拡張させていくといいと思います。内部コマンドの増設は，メモリの関係上難しいものがありますが，このようにしていけば，ディスクの容量があるかぎりコマンドが作れますからね。

ちなみに僕は，アセンブラで開発をすることが多く，いちいちエディタやアセンブラをロードするのはめんどくさかったので，モニタ上からエディタやアセンブラを一発起動させるコマンドを作ってしまいました（ごくたまに暴走することもありますけどね）。そして，空いているG-RAMをエディタとアセンブラ専用のキャッシュディスクとして使っています。

1回アセンブラやエディタを立ち上げると，2回目からはディスクアクセスなしでG-RAMから立ち上がるので，とっても気持ちがいい。これのおかげで，ずいぶんとプログラムの開発効率も上がりました。ぜひ皆さんも，自分にあったシステムの改造をしてみませんか？　楽しいですよ〜。

リスト2

```
 1
 2 ;TYPE COMMAND Ver 1.0
 3
 4 #MPRINT EQU $1FE2
 5 #GETL   EQU $1FD3
 6 #ERROR  EQU $2033
 7
 8         ORG   $9000
 9 BEGIN
10         CALL  #MPRINT
11         DB    "*** TYPE Ver 1.0 ***"
12         DB    $0D,00
13         CALL  #MPRINT
14         DB    "File Name:",00
15         CALL  #GETL
16         LD    A,(DE)
17         CP    $1B
18         RET   Z
19         LD    HL,10
20         ADD   HL,DE
21         EX    DE,HL
22         CALL  FILEREAD
23         JP    C,#ERROR
24         CALL  TYPEMAIN
25         RET
26 BEGINEND
```

リスト3

```
 1
 2 ;FILE READ ROUTINE
 3
 4 #RDD     EQU $1FA6
 5 #FILE    EQU $1FA3
 6 #ROPEN   EQU $2009
 7 #DTADR   EQU $1F70
 8
 9          ORG   BEGINEND
10
11 FILEREAD
12          LD    A,04
13          CALL  #FILE
14 FR2
15          CALL  #ROPEN
16          RET   C
17          LD    HL,MAINEND
18          LD    (#DTADR),HL
19          CALL  #RDD
20          RET
21 READEND
22
```

（ディスク専用です）

リスト4

```
 1
 2 ;TEXT PRINNT ROUTINE
 3
 4 #PRINT   EQU $1FF4
 5 #LTNL    EQU $1FEE
 6 #PAUSE   EQU $1FC7
 7
 8          ORG   READEND
 9
10 TYPEMAIN
11          LD    HL,MAINEND
12 TM2
13          LD    A,(HL)
14          OR    A
15          JR    Z,TM3
16          INC   HL
17          CALL  #PRINT
18          CALL  #PAUSE
19          DW    TM3
20          JR    TM2
21 TM3
22          CALL  #LTNL
23          RET
24 MAINEND
```

THE SENTINEL

●リロケータブルアセンブラ登場

さて，すでに先月の「リロケータブルフォーマットの取り決め」で概要をお知らせしたとおり，S-OS上でリロケータブルファイルを扱うためのアセンブラ，WZDを発表します。予告ではこのアセンブラが出力したファイルをリンクするリンカ，WLKも同時発表となっていましたが，ページ数の都合によりアセンブラ部分のみの発表となりました。あらかじめご了承ください。

今回のWZDはエディタで書かれたソースプログラムをリロケータブルファイルに変換するためのものです。これをさらに実行形式にするには来月掲載予定のWLKが必要です。ページが許せば専用のライブラリアン，WLBも来月（以降）で紹介することになります。

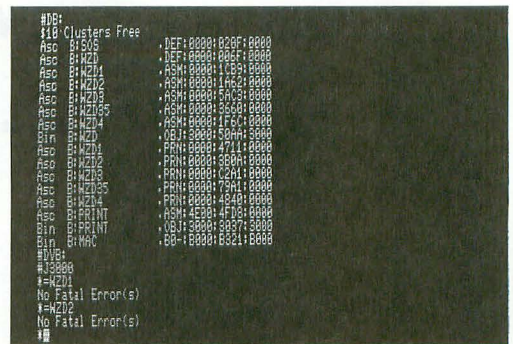
以前から何度か紹介しているように，これらのシステムはもともとS-OS上でC言語を効率よく運用するために開発されたものです（もちろんC言語を動かす以外にもおいしい使い方はたくさん考えられますが)。C言語が加わればS-OSの広がりもさらに大きくなるでしょう。S-OS上にどのようにしてC言語を持ってくるかは，また今後のお楽しみということにしておきましょう。

第96部
リロケータブルアセンブラ WZD

●リロケータブルファイルの扱い

今回のWZDを使用する際には必ずフロッピーディスクまたはRAMディスクが必要になってきます。これまでS-OSの世界ではディスクがないと使えないアプリケーションというのはごく限られた存在でした。ディスクエディタや変身セットの一部の機能，SLANGなどのインクルード機能，ファイル入出力ライブラリといったところでしょうか。

S-OSシステムはカセットテープでもディスクでも同様のファイルを扱うというポリシーを堅持していても，システムに対する要求が次第に高度なものになってくると，どうしても1文字単位の入出力やリロケータブルファイルのリンクなどといったカセットテープの不得意な問題が浮かび上がってきます。編集部でもそれらをカセットテープでも変わらずに実現するための方法をいろいろ論議したものでした。カセットベースでのダイナミックリンクの可能性などは非常に興味深いテーマです。しかし，いつも結論は「技術的に不可能ではないが，かなり不毛」というところに落ち着きます。

今回のシステムはS-OSにおけるひとつの岐路といえるかもしれません。

●S-OSの系譜(11)

パソコンユーザーにとってもっとも親しみのあるプログラミング言語，それはやはりBASICでしょう。1986年9月，ついにS-OS上にBASIC言語が登場しました。このFuzzyBASICとはいったいいかなるものでしょうか。単に「パソコンユーザーの作った整数型BASIC」だと思っていると大間違いです。

ブロックIFなどの構造化命令は当然として，ユーザー定義手続き，ユーザー定義関数（もちろん局所変数つき)，I/O配列，ユーザースタック，Hu BASICでお馴染みのKEY0命令，はてはブロック転送命令まで備えています。BASICという言語の持つ泥くさいイメージとは違い，洗練された高級言語としてプログラミングユーザーに受け入れられました。

確かにプログラミングに適した言語仕様，省略形のサポートやシャープ，マイクロソフト系両方に対応したコマンドなどのユーザーインタフェイス部分も備えたフレンドリな面，かなり高速なうえ，使う側が心得ていれば，高速アクセス変数，マシン語に密着した命令群を使ってさらに最適化することができるなどのマニアックな面も魅力ですが，BASICという言語にこだわりつつも，従来のパソコン用BASICを超える発想とこだわり方，そしてユニークさがなにより魅力的だったのではないでしょうか。

その後，FuzzyBASICはグラフィックのサポート，命令の強化などを加え，やがて現れるFuzzyBASICコンパイラによってさらに完成度を高めていくことになります。

また，直接全機種共通システムとは関係ありませんが，1986年9月号は高速グラフィックパッケージMAGICという弩級プログラムも掲載され，両プログラムのためにいまでは過去最高の希少バックナンバーとして知られています。MAGICはその高性能もさることながら，グラフィック共通化の鍵として以後S-OSシステムと密接な関係を持ちつつ，独自の世界を広げていきました。

全機種共通S-OS"SWORD"要

リロケータブルアセンブラ WZD

Ishigami Tatsuya
石上 達也

6月号で予告したリロケータブルアセンブラ，WZDの登場です。リロケータブルフォーマットによってS-OSの新しい可能性が広がっていきます。FDDがないと使用できませんので注意が必要です。なお，WZDは"ダブルゼーダ"と読んでください。

先月号の予告ではリンカとともに発表することになっていましたが，ページの都合で，リンカの発表は来月号で，ということにさせていただきます。とりあえず，リロケータブルアセンブラ WZD（ダブル・ゼーダと読んでください）を発表させていただきます。それから先月号では触れるのを忘れてしまいましたが，このWZDはフロッピーディスクがないと動作しません。

このプログラムは，いつかも予告したようにS-OS上でCコンパイラを走らせるために開発しましたが，単体でも十分実用的だと信じています。ちなみに，WZDとWLKの原形はCP/M上で作りましたが，そこからはWZD自身を使って開発しました。来月号(？)で発表予定のWLBは90%以上WZDを使って作成しました。

私の学部1年のほとんどの時間をかけて作ったプログラムです。なるべく多くの人に使ってもらいたいのでこのプログラムはPDSとします（どっかのネットに入れた場合は一応私に知らせてください）。

では使い方なんぞを

WZDの起動の仕方には，2種類あります。S-OSのコマンドラインからパラメータを指定する方法と，WZD内のコマンドラインから，パラメータを指定する方法です。

●S-OSのコマンドラインから

まず，S-OSの拡張をしている方は，

　#　WZD　パラメータ

とすると動作を開始します。アセンブルが終わると#を示して，S-OSのコマンドラインに戻ります。

次に，S-OSの拡張をされていない方は，

　#LWZD

で，WZDをディスクから読み込んで，

　#J3000　パラメータ

とすると動作を開始します。あとは拡張をしているときと同じです。この方法は，1本のプログラムをアセンブルしたいときに用いると便利です。

●WZDを起動してから

　#LWZD
　#J3000

でWZDを起動すると入力を促すプロンプト'*'を表示してWZDのコマンドラインに入ります。'*'の後ろに続けてパラメータを与えます。パラメータによって指示された動作が終了すると，再び入力を促すプロンプト'*'を表示して，WZDのコマンドラインに入ります。

さらに，別のプログラムをアセンブルしたい場合には，1回目と同じようにパラメータを与え，WZDを終了したいときには，シフト+ブレイクキーを押すとS-OSのコマンドラインに戻ります。この方法は，アセンブルしたいプログラムが複数あるときに便利です。

パラメータといったって

基本的には，MACRO-80に準じています。以下に，WZDが起動してからパラメータを指定した場合について具体的な使い方の例を挙げておきます。

パラメータの指定で一番オーソドックスなのは，

　*[リスト・ファイル名],[リロケータブルファイル名]=[ソースファイル名]

のかたちです。

で，たいていは，[リロケータブルファイル名]と[ソースファイル名]は拡張子が違うだけであとは同じ，[リストファイル名]は無指定（つまりリストファイルは作らない）にしたいわけで，そんなときには，

　*=［ファイル名］

とすれば，タイプ数も少なくてすみます。やはり，[リストファイル]も同じ名前ではしいというときには

　*=［ファイル名］/L

です。

ソースファイルはBドライブにあって，リロケータブルファイルはAドライブに作りたいときは，

　*A:[リロケータブルファイル名]=B:[ソースファイル名]

です。さらに詳しくはリファレンスマニュアルを見てください。

プログラムについて

ニーモニックからマシン語への変換については，1989年2月号掲載のREDAのルーチンをほぼそのまま使わせていただいてます。ただREDAのルーチンは，裏レジスタまでも駆使し，テキストポインタにDEレジスタを用いたりと，かなりスピードを重視したルーチンが多かったのですが，WZDではスピードはあまり重視せずに，安全性を重視して一部書き直してあります。

さらに，ラベルの検出方法をREDAのオープンハッシュ法からチェーンハッシュ法に切り替えています。別に深い意味はありませんが，心持ち速くなったような気がします。ハッシュ法とはなにかを説明しよう

使用上の注意

EQU文のパラメータは1パス目のその時点において，値が確定している必要があります。このことによって一部，REDAのソースファイルがアセンブルできないことがあるかもしれませんが，そのときは値が確定する順にラベルを並べ替えてください。

PHASE文とDEPHASE文のあいだにおいてセグメントのモードを切り替えてはいけません。また，PHASE文とDEPHASE文のあいだではアドレス情報がアブソリュート値で処理されます。

インクルードファイルのネスティングは4重までです。呼ばれたファイルから呼び出したファイルを，呼び出しては絶対にいけません（無限ループに陥ります）。

リストファイル中のデータの中に，**マークが現れることがありますが，これはその値がまだ確定していないことを示しています。具体的にはアブソリュート値で処理されていないラベルを用いたときなどに現れます。

かと思ったのですが，近いうちにZ80's Barのほうで詳しくやるようなので，そちらを見てください。

また，WZD1.ASMは，最初Small-Cで書いておいてその後，私がASMファイルを書き直したものです。ラベルにCCxxというのが多く使われているのは，そのためです。この方法は，いきなりマシン語のコーディングをしなくてすむし，かといってフローチャートのようなものを細かく書かなくていいのでたいへん便利です。

ただニーモニックからマシン語への変換をくそ真面目に行うプログラムなので，特にテクニックのようなものはそんなにありません。ひたすら，腕力の勝負でした。

このようなプログラムを作るコツは，それぞれのルーチンをできるだけ一般化してしまうということです。

たとえば，パラメータは，HLレジスタ，DEレジスタ，BCレジスタの順に用いて戻り値は，HLレジスタに代入して返すとか，原則的に各ルーチンは，戻り値を代入するレジスタ以外は保存しておくとか，決めておくといろいろと便利です。それから，ポインタはなるべくあと戻りさせないというのもわかりやすいプログラムを書くうえでけっこう有効です。

このプログラムを作っていて気がついたものとしては，すべてのレジスタを破壊しないで，ある（メモリ上に取ってある）スイッチの値を調べるときには，

```
PUSH    HL
LD      HL，スイッチのアドレス
LD      H,(HL)
INC     H
DEC     H
POP     HL
        ;ここでスイッチの値について（0かそれ以外か）Zフラグが変化する
```

と方法があるということです。1パス目か2パス目かの判断に多用しています。

ファイル関係のサブルーチンを今回は自作したのですが，なるべくCP/MのFCB(File Control Blockの略だってさ）に似せようと思ったのであまり効率のよいものではありません。が，このようなサブルーチンを作成するとき，2カ所以上で無関係に同じファイルに書き込みを行おうとした場合の処理を，見落としがちなので少しばかり触れておきます（以前発表されたSLANG用ファイル入出力サブルーチンもアプリケーションのほうで処理してやらなくてはならないようです）。

まず，システムが使用していないファイル属性からひとつの値を選びそれを「書き込み中」を表す属性にしてやります（今回は5にしました）。

ファイルを書き込み用にオープンしたときは，ファイルに対してその属性を与えてやります。こうしてやれば，次の書き込み用ファイルをオープンする前にそのファイルの属性は「書き込み中」を表していないかどうか，調べるだけですみます。特にS-OSの場合は，わざわざ調べなくてもファイルのオープン時に，Bad File Modeエラーが発生します。

そして，クローズするときに元々の属性に戻してやれば，(ASCIIファイルなら4とか，バイナリファイルであれば1とか）ちゃんとディスクに収まるのです。

WZD をソースリストからアセンブルするときは，

```
#LWZD
*=WZD1
*=WZD2
*=WZD3
*=WZD35
*=WZD4
*[ここで，シフト+ブレイクを押す]
```

以上により，WZD1.REL，WZD2.REL，WZD3.REL，WZD4.RELが，作成されますので，次にこれらのファイルを，

```
# WLK
*/P:3000，/D:6000
*WLK1, WLK2, WLK3, WLK4, WLK/N:P
```

としてリンクすれば，ここに掲載されているリストと同じものがWZD.OBJというファイルに得られます。

プログラムにはアルゴリズムを表している部分とデータを表している部分がありますが（特にマシン語では露骨にそうなる），ときとして両者を分けたいときがあります。アブソリュートアセンブラならワークエリアを後ろに持っていくとかすればよいのですが，複数のリロケータブルファイルからオブジェクトファイルを作る場合はそうはいきません。そこでリンカに複数のPCを持たせてアルゴリズムを表すコードがきたら，コードセグメントを受け持つPCを用いて，データの部分はデータセグメントを受け持つPCを用いて処理してやります。

なお，WZDでは，コードセグメントとデータセグメントのほかにワークセグメントが，用意されています。MACRO-80で用意されていたアブソリュートセグメントは，用意されていません。具体例はWZD自身のソースプログラムを見てください。

最後に

この記事の内容がよくわからなかったとしても，それは気にするほどのことではないと思います。WZDを使っているうちに，なんとなくわかってくるから，きっと。なお質問点があれば質問箱にお寄せください(初心者・女性大歓迎，ひやかし・オタク小歓迎)。ひょっとすると，来々月号でWZDに関する質問特集をやれるかもしれません。

それから，私に逆ポーランドから普通の数式に変換するアルゴリズムを教えてくださった宝塚市のTAOさん，熊本市の茶円亮さん，PurePASCALの藤木健士さん，TED-750の鈴木典雄さんありがとうございました。

「材料の力学」という科目の単位を修得しだい，逆アセンブラの作成に取り掛かろうと思います。でも，なんかコンパイラをその前にひとつ作りたいな，んん，その前に単位が先かなぁ？

では，来月のWLKとWLBの発表でお会いしましょう（それまで，16ビット機に乗り移っちゃ駄目だよ！）。

セグメントについて

セグメントの異なるエリアに，JR命令やDJNZ命令でジャンプすることはできません（JR命令やDJNZ命令のオペランドの値は，WZD内で処理を完了する必要がある。その処理をリンカに持ち越してはいけない)。

コードセグメントとデータセグメントは，アドレスを，重ねてはいけませんが，ワークセグメントは，コードセグメントおよびデータセグメントと重ねてかまいません。

また，ワークセグメントの内容は，リンカにアドレス情報を伝えるだけで，実体は作成することができません。さらに例外的にこのセグメントだけPCのあと戻りができます。

いったんORG命令を処理すると同一ファイル内においてはそのセグメントのアドレス情報は，すべてアブソリュート値で処理されます(MACRO-80でいうところの，ASEGです)。そのようなときには，前述の理由により，ORG命令を挟んで相対ジャンプはできません（絶対ジャンプはできます)。

S-OSでは，64Kバイトを超える大きさのファイルを扱うことはできません。ソースファイルの大きさが64Kバイトを超えることはまずないと思いますが（だってエディタが扱えないでしょ)，30Kバイトくらいのソースファイルからリストファイルを作った場合，64Kバイトを超えてしまう可能性がありますので注意してください。そのときはWZDは，“Bad Record”と，エラーメッセージを表示してすべてのアセンブル作業を中止します。

表1 リファレンスマニュアル

パラメータ

基本的には,

[リロケータブルファイル名], [リストファイル名]=

[ソースファイル名]

の書式でパラメータを与えるが, 続けて以下のスイッチを併用することによってパラメータの一部またはソースファイルの一部を省略することができる。

なお, [リストファイル名], [リロケータブルファイル名], [ソースファイル名] において拡張子名を省略した場合にはそれぞれ, '.PRN', '.REL', '.ASM'になります。

/L

リストファイルのファイル名をソースファイルのファイル名と同じにする。

/R

リロケータブルファイルのファイル名をソースファイルのファイル名と同じにする。

/K

ソースファイルには, シフトJISコードで表される漢字などが含まれていることを示す。具体的には, 以下のようになります。

=[ファイル名]

ソースファイルとリロケータブルファイルを [ファイル名] にする。

,=[ファイル名]

ソースファイルを [ファイル名] にする。このときリロケータブルファイルとリストファイルは作らない。ソースファイル中のアセンブルエラーとなる箇所をチェックするのに便利。

=[ファイル名] /L

ソースファイル, リロケータブルファイル, リストファイルすべてを [ファイル名] とする。

,=[ファイル名] /L

ソースファイルとリストファイルを [ファイル名] にする。

[ファイル名1]=[ファイル名2]

ソースファイルを [ファイル名2] に, リロケータブルファイルを [ファイル名1] にする。

[ファイル名1]=[ファイル名2] /L

ソースファイル, リストファイルを [ファイル名2] に, リロケータブルファイルを [ファイル名1] にする。

,[ファイル名1]=[ファイル名2]

ソースファイルを [ファイル名2] に, リストファイルを [ファイル名2] にする。

,[ファイル名1]=[ファイル名2]/R

ソースファイル, リロケータブルファイルを [ファイル名2] に, リストファイルを [ファイル名1] にする。

疑似命令

REDAに比べて以下の命令が拡張されています。

END [式]

アセンブル作業を終了します。この行の後ろになにを書いてもアセンブラには無視されます。式がある場合にはその値を実行開始アドレスに指定します ([式] は省略可能)。

.KANJI

以後のソースプログラムはシフトJISコードで書かれていることを示します (パラメータのスイッチで/Kを指定した場合と同じ) 漢字を用いるときには, 先立ってこの命令も書いておいてください。

.NKANJI

以後のソースプログラムはASCIIコードのみで書かれていることを示します。先の.KANJI命令と組み合わせることによって, REDAにおけるKスイッチの働きをアセンブル中に切り替えられます。

CSEG

以下のプログラムはコードセグメントに配置されることを示します。

DSEG

以下のプログラムはデータセグメントに配置されることを示します。

WSEG

以下のプログラムはワークセグメントに配置されることを示します。

EXT [ラベル名]

[ラベル名] で表されたラベルはこのファイルにはなく, 他のファイルで外部ラベルとして宣言されていることを示します。同じファイル中に同じ名前があった場合は, Multi Defined エラーとなります。

PUBLIC [ラベル名]

[ラベル名] で表されたラベルを外部 (参照可能) ラベルとして宣言します。同じファイル中に [ラベル名] で表されたラベルがない場合は, Undefined Labelエラーとなります。

[ラベル名]##

[ラベル名] で表されたラベルは, 外部ラベルとして処理されます。なお, この表記は式中においてのみ可能です。

[ラベル名]::

[ラベル名] で表されたラベルを, 現時点でのPCの値を表す内部ラベルとして宣言するとともに, 外部 (参照可能) ラベルとしても宣言します。この表記は, ソースプログラム中の1列目においてのみ可能です。

.PHASE [式]

以下のプログラムのPCを一時的に [式] の値にします。ただ, PCを一時的に変更するだけで, オブジェクトプログラムは以前のアドレスと連続して配置されます (プログラムの転送は行われませんので, プログラマの責任において行ってください)。

.DEPHASE

PHASE文で一時的に変更したPCを元の値に戻します (元の値といってもPHASE文とDEPHASE文のあいだのプログラム分だけPCは進んでいます)。

INCLUDE [ファイル名]

ソースプログラム中のこの位置に [ファイル名] で示されたファイルをそっくり読み込みます。読み込まれたファイルの中でさらにINCLUDE文が使われていても (つまりINCLUDE文がネスティングしていても) 4重までなら構いません。

PAGE [式]

リストファイルにおいて1ページあたりの行数を指定します。ページごとに区切られたくない場合は, PAGE 0FFFFHとしておいてください。現実的には区切られません (6万行以上からなるファイルは存在しない, と思う)。デフォルトの [式] の値は0FFFFHです。

TITLE [文字列]

リストファイルにおいてこのプログラムのタイトルを [文字列] にします。ここで, タイトルとはリストファイルの一番最初の行にプリントされるタイトルであって, リストファイルのファイルネームとは関係ありません。

表2 エラーメッセージ

Syntax Error

文法エラーが発生した。

Undefined Label Error

未定義なラベルが使用された。

Redefinition Error

同じラベルを2度以上定義しようとした。

Illegal Label Error

ラベル名がおかしい。

Illegal Opecode Error

オペコードがおかしい。

Illegal Operand Error

オペランドがおかしい。

Too Many Labels Error

使用されたラベルが多すぎる (たいていの機種では, まず起こらない)。

Missig Label Error

ラベルのあるべきところにラベルがない。

Too Far Error

JRやDJNZ命令などでオペランドで表された値が届く範囲にない。

Illegal Expression Error

式の表し方がおかしい。

Missing [ )] Error

")"の文字がない。

Missing [.] Error

"."の文字がない。

Missing Quote Error

文字列の表し方がおかしい。

Illegal ORG Error

ORG命令のオペランドがおかしい (アドレスの低いほうに指定しなおしたなど)。

Value Error

1パス目で決まっていなければならない値が決まっていない (EQU, ORG命令など)。

Relocation Error

JRまたは, DJNZ命令のオペランドにおいて, 値が不確定な意味のない式を書いた (JR Label *2など)。

Internal Error

WZDの内部で起きたエラー (このエラーがもし発生したらそれは, WZD内にバグが潜んでいるということです。そのときは, Oh!X編集部経由で私に知らせてください)。

リスト1 WZDダンプリスト

```
3000 ED 73 00 60 ED 7B 6A 1F : B1
3008 ED 5B 76 1F 06 01 1A 13 : 11
3010 A7 28 07 FE 20 20 F7 04 : 0F
3018 18 F4 78 32 02 60 AF 32 : F9
3020 57 60 32 9F 60 32 B1 60 : 2B
3028 32 A3 61 32 DB 61 3A 02 : E0
3030 60 FE 01 20 05 CD 6C 31 : EE
3038 18 0C 2A 76 1F 7E 23 FE : 82
3040 20 20 FA 22 03 60 21 88 : 68
3048 31 CD E1 33 11 57 60 2A : 04
3050 03 60 CC 13 34 11 B1 60 : 98
3058 2A 03 60 CD 13 34 22 03 : C6
3060 60 21 8A 31 CD E1 33 11 : 2E
3068 9F 60 2A 03 60 CC 13 34 : 9F
3070 22 03 60 21 88 31 CD E1 : 0D
3078 33 11 57 60 2A 03 60 CC : 54
-----------------------------------
SUM: 6C DC 25 00 AE B7 6B 00 ECCF

3080 13 34 22 03 60 2A 03 60 : 59
3088 7E A7 CA D1 30 21 8C 31 : CE
3090 CD E1 33 28 08 21 8F 31 : F2
3098 CD E1 33 20 0C 11 B1 60 : 2F
30A0 21 57 60 CD 13 34 C3 85 : 34
30A8 30 21 92 31 CD E1 33 28 : 1D
30B0 08 21 95 31 CD E1 33 20 : F0
30B8 0C 11 9F 60 21 57 60 CD : C1
30C0 13 34 C3 85 30 21 A4 31 : B5
30C8 CD 4F 34 CD 5A 34 C3 85 : F3
30D0 30 3A 57 60 A7 11 57 60 : 90
30D8 21 98 31 C4 F5 33 3A B1 : C1
30E0 60 A7 11 B1 60 21 9C 31 : 17
30E8 C4 F5 33 3A 9F 60 A7 11 : DD
30F0 9F 60 21 A0 31 C4 F5 33 : DD
30F8 11 57 60 21 9F 60 CD 26 : DB
-----------------------------------
SUM: 95 EF BC CD 67 08 55 1E 4E99

3100 34 21 C0 31 D4 38 34 11 : 97
3108 57 60 21 B1 60 CD 26 34 : 10
3110 21 C0 31 D4 38 34 11 9F : 02
3118 60 21 B1 60 CD 26 34 21 : DA
3120 C0 31 D4 38 34 3A 57 60 : 22
3128 A7 21 C0 31 CC 38 34 CD : BE
3130 02 32 CD 47 32 2A 3D 62 : 43
3138 7C B5 F5 C4 19 48 F1 C4 : 5D
3140 58 31 CC 4F 34 21 5B 31 : 85
3148 CD 4F 34 3A 02 60 FE 01 : EB
3150 CA 1E 30 ED 7B 00 60 C9 : A9
3158 4E 6F 00 20 46 61 74 61 : 59
3160 6C 20 45 72 72 6F 72 28 : BE
3168 73 29 0D 00 3E 2A CD F4 : D2
3170 1F ED 5B 76 1F CD D3 1F : BB
3178 1A FE 1B CA 5A 34 13 1A : B8
-----------------------------------
SUM: 46 DC 11 D2 A4 BF AA 66 DFA2

3180 A7 28 E9 ED 53 03 60 C9 : 24
3188 3D 00 2C 00 2F 72 00 2F : 39
3190 52 00 2F 6C 00 2F 4C 00 : 68
3198 41 53 4D 00 52 45 4C 00 : C4
31A0 50 52 4E 00 55 73 61 67 : 80
31A8 65 3A 20 57 5A 44 20 5B : 2F
31B0 52 45 4C 5D 2C 5B 50 52 : 69
31B8 4E 5D 3D 41 53 4D 0D 00 : D6
31C0 53 6F 75 72 63 65 20 66 : F7
31C8 69 6C 65 20 6E 61 6D 65 : FB
31D0 20 69 73 20 6E 6F 74 20 : 8D
31D8 67 69 76 65 6E 0D 00 41 : 67
31E0 53 4D 09 66 69 6C 65 20 : 69
31E8 3A 00 0D 52 45 4C 20 66 : B0
31F0 69 6C 65 20 3A 00 0D 50 : F1
31F8 52 4E 20 66 69 6C 65 20 : 80
-----------------------------------
SUM: 57 5D E6 A3 00 AE CE 2E 3964

3200 3A 00 AF 32 2E 62 32 4F : 2C
3208 62 CD EC 44 CD 4B 44 21 : DC
3210 00 00 22 55 60 3E 04 11 : 2A
3218 57 60 21 C3 60 CD D7 4C : EB
3220 21 57 60 DA 71 33 3E FF : 93
3228 32 D9 4E CD 90 33 38 0B : 2C
3230 3A 4F 62 A7 20 05 CD E2 : 66
3238 37 18 F0 3A 55 60 3D 32 : 9D
3240 55 60 3C C2 2B 32 C9 3E : 17
3248 01 32 2E 62 AF 32 4F 62 : 55
3250 32 55 60 CD 67 44 3E 04 : A1
3258 11 57 60 21 C3 60 CD D7 : B0
3260 4C 21 57 60 DC 71 33 3E : E2
3268 FF 32 D9 4E 3A B1 60 A7 : 4A
3270 28 15 3E 01 11 B1 60 21 : BF
3278 DB 61 CD C0 4D 21 B1 60 : 48
-----------------------------------
SUM: 9E CB 43 97 A9 7F 98 CC E129

3280 DA 71 33 AF 32 48 4F 3A : 30
3288 9F 60 32 3F 62 A7 28 15 : B6
3290 3E 04 11 9F 60 21 A3 61 : 77
3298 CD C0 4D 21 9F 60 DA 71 : 45
32A0 33 AF 32 74 4F CD 90 33 : 67
32A8 38 0B 3A 4F 62 A7 20 05 : FA
32B0 CD E2 37 18 F0 3A 55 60 : DD
32B8 3D 32 55 60 3C C2 A5 32 : F9
32C0 3A B1 60 A7 28 14 CD 84 : 7F
32C8 35 3E FF CD B5 33 21 00 : 48
32D0 BC 22 25 50 21 DB 61 CD : 7D
32D8 3B 4E 3A 9F 60 A7 C8 3E : 6F
32E0 00 CD C8 33 21 00 BD 22 : C8
32E8 25 50 21 A3 61 CD 3B 4E : F0
32F0 C9 ED 53 6D 33 3A 55 60 : 98
32F8 3C 32 55 60 FE 04 38 09 : 66
-----------------------------------
SUM: 89 FE 0A EF 81 B4 3A 53 AABA

3300 21 59 33 CD 4F 34 C3 5A : 1A
3308 34 2A 55 60 01 12 00 CD : F3
3310 46 48 11 57 60 19 22 6F : 00
3318 33 ED 5B 6D 33 1A CD 3F : 41
3320 34 38 05 13 77 23 18 F5 : 2B
3328 36 00 ED 53 6D 33 01 38 : 4F
3330 00 2A 55 60 CD 46 48 11 : 4B
3338 C3 60 19 3E 04 ED 5B 6F : 35
3340 33 CD D7 4C 2A 6F 33 DA : C9
3348 71 33 2A 55 60 29 11 D9 : 96
3350 4E 19 36 FF ED 5B 6D 33 : 84
3358 C9 49 4E 43 4C 55 44 45 : CD
3360 20 6C 65 76 65 6C 20 6F : C7
3368 76 65 72 20 0D 00 00 00 : 7A
3370 00 E5 21 82 33 CD 4F 34 : 0B
3378 E1 CD 4F 34 CD EE 1F C3 : CE
-----------------------------------
SUM: 2D 5F 20 24 CD 71 F1 13 F59D

3380 5A 34 43 61 6E 20 6E 6F : 9D
3388 74 20 6F 70 65 6E 20 00 : 66
3390 06 50 21 05 60 C5 E5 CD : 53
3398 AA 4E E1 C1 A7 28 0B FE : 72
33A0 0D 28 0B 77 23 05 20 ED : EC
33A8 B7 C9 AF 77 37 C9 3E 0D : F1
33B0 77 23 AF 77 C9 E5 21 B1 : 40
33B8 60 6E 2C 2D E1 C8 CD 1E : BB
33C0 4F D0 CD 33 20 C3 5A 34 : 90
33C8 E5 B7 21 3F 62 6E 2C 2D : 25
33D0 28 07 D5 C5 CD 4A 4F C1 : F0
33D8 D1 E1 D0 CD 33 20 C3 5A : BF
33E0 34 ED 5B 03 60 7E A7 28 : 2C
33E8 07 1A BE C0 13 23 18 F5 : E2
33F0 ED 53 03 60 C9 06 0D 1A : 99
33F8 CD 3F 34 38 0A 05 28 07 : B6
-----------------------------------
SUM: 3B 7C 2C 88 A6 3D 56 BD 7A89

3400 FE 2E 28 03 13 18 F0 1A : 8C
3408 FE 2E C8 3E 2E 12 13 CD : 52
3410 13 34 C9 06 0D AF 12 7E : 62
3418 CD 3F 34 D8 05 C8 7E 23 : 86
3420 12 13 AF 12 18 F1 1A A7 : B0
3428 37 C8 7E A7 37 C8 1A A7 : E4
3430 C8 13 BE 23 37 C0 18 F6 : C1
3438 CD 4F 34 CD 5A 34 C9 FE : 72
3440 3D 28 0A FE 2F 28 06 FE : C8
3448 2C 28 02 A7 C0 37 C9 7E : 3B
3450 23 A7 C8 E5 CD F4 1F E1 : 38
3458 18 F5 3A DB 61 A7 21 00 : 4B
3460 BC 22 25 50 21 DB 61 C4 : 74
3468 3B 4E 3A A3 61 A7 21 00 : 8F
3470 BD 22 25 50 21 A3 61 C4 : 3D
3478 3B 4E ED 7B 00 60 C9 CD : E7
-----------------------------------
SUM: 4D D8 8B EB F3 CD 63 7C 97D1

3480 97 44 E5 21 2E 62 6E 2D : 0C
3488 E1 C0 F5 3A 9F 60 A7 28 : 9E
3490 08 F1 F5 CD 28 37 CD 62 : 49
3498 37 3A 52 62 FE 60 20 20 : C3
34A0 F1 D5 E5 ED 5B 50 62 16 : BB
34A8 00 21 53 62 19 77 7B E1 : C2
34B0 D1 3C E6 1F 32 50 62 C0 : B6
34B8 C5 06 20 CD 6A 35 C1 C9 : E1
34C0 C5 3A 50 62 47 3A 52 62 : E6
34C8 A7 C4 6A 35 C1 3E 60 32 : 9B
34D0 52 62 3E 01 32 50 62 F1 : C8
34D8 32 53 62 C9 C5 06 01 FE : 7A
34E0 84 28 10 FE E4 28 0C 06 : D8
34E8 02 FE 82 28 06 FE E5 28 : BB
34F0 02 18 0E F5 3A 9F 60 A7 : FD
34F8 C4 6A 37 CD 97 44 10 FB : 18
-----------------------------------
SUM: 7A C2 90 0E BD 7C 78 AA F33F

3500 F1 C1 E5 21 2E 62 6E 2D : E3
3508 E1 C0 F5 C5 CD 84 35 C1 : A2
3510 F1 CD 90 35 C9 CD DC 34 : 29
3518 F5 7D CD 02 35 7C CD 02 : C1
3520 35 F1 C9 CD 97 44 E5 21 : 9D
3528 2E 62 6E 2D E1 C0 E5 21 : D2
3530 52 62 BE E1 20 1A D5 E5 : 47
3538 3A 50 62 16 00 5F 21 53 : D5
3540 62 19 71 E1 D1 3C E6 07 : C7
3548 32 50 62 C0 06 08 18 1A : E4
3550 F5 3A 50 62 47 3A 52 62 : 16
3558 A7 C4 6A 35 F1 32 52 62 : E1
3560 3E 01 32 50 62 79 32 53 : 21
3568 62 C9 04 05 C8 3A 52 62 : EA
3570 80 3D CD 90 35 D5 E5 21 : 2A
3578 53 62 7E CD 90 35 23 10 : F8
-----------------------------------
SUM: 4A A0 9C F8 8F 19 3A 69 CCC8

3580 F9 E1 D1 C9 3A 50 62 47 : A7
3588 CD 6A 35 AF 32 50 62 C9 : C8
3590 E5 D5 C5 F5 CD B5 33 F1 : 1A
3598 C1 D1 E1 C9 E5 21 2E 62 : D2
35A0 6E 2D E1 C0 D5 E5 F5 3A : 25
35A8 51 62 6F 3C 32 51 62 26 : 69
35B0 00 11 73 62 19 F1 77 E1 : 48
35B8 D1 C9 CD 9C 35 F5 7D CD : 77
35C0 9C 35 7C CD 9C 35 F1 C9 : A5
35C8 3A 51 62 21 73 62 A7 C8 : 52
35D0 F5 7E 23 CD 02 35 F1 3D : C8
35D8 18 F4 3A A5 62 E6 01 20 : 54
35E0 09 79 CD 7F 34 78 CD 7F : C6
35E8 34 C9 CD C8 35 3E 82 CD : 54
35F0 DC 34 C9 3A A5 62 E6 01 : 01
35F8 79 CA 7F 34 CD C8 35 3E : FE
-----------------------------------
SUM: 71 92 59 45 C1 24 64 EA BA5D

3600 84 CD DC 34 C9 3A 2E 62 : F4
3608 A7 C8 3A 9F 60 A7 C8 AF : C6
3610 32 13 62 3E 01 32 14 62 : 8E
3618 C3 79 36 3A 2E 62 A7 C8 : AB
3620 3A 9F 60 A7 C8 AF 32 13 : 9C
3628 62 CD 62 37 CD 3F 37 CD : D8
3630 62 37 CD 62 37 C9 3A 2E : 30
3638 62 A7 C8 3A 9F 60 A7 C8 : 79
3640 E5 21 20 20 22 16 62 22 : 02
3648 18 62 22 1A 62 3E 07 32 : 8F
3650 13 62 E1 C9 F5 C5 D5 E5 : 93
3658 11 15 62 2A 13 62 26 00 : 4D
3660 19 77 23 36 00 3A 13 62 : 98
3668 3C 32 13 62 FE 16 D4 76 : 41
3670 36 E1 D1 C1 F1 C9 CD 89 : B9
3678 36 CD 62 37 CD 5C 3A CD : CC
-----------------------------------
SUM: 62 BC F3 82 0B 7C 4D 78 2F62

3680 3F 37 CD 4A 37 CD 62 37 : 2A
3688 C9 3A 2E 62 A7 C8 3A 9F : DB
3690 60 A7 C8 2A A8 62 7C B5 : 34
3698 CC DC 36 2A A8 62 23 22 : 57
36A0 A8 62 2B D5 ED 5B A6 62 : 5A
36A8 B7 ED 52 D1 38 06 21 00 : 26
36B0 00 22 A8 62 CD 00 37 21 : 51
36B8 15 62 7E A7 28 06 23 CD : BA
36C0 C8 33 18 F6 3A 14 62 A7 : 60
36C8 28 05 CD 86 37 18 05 3E : 12
36D0 0D CD C8 33 AF 32 13 62 : 2B
36D8 32 14 62 C9 21 00 00 22 : B4
36E0 A8 62 3A AA 62 A7 C8 21 : E0
36E8 AA 62 7E 23 A7 28 05 CD : 4E
36F0 C8 33 18 F6 3E 0D CD C8 : E9
36F8 33 21 01 00 22 A8 62 C9 : 4A
-----------------------------------
SUM: 24 F8 7C EA F2 A2 D2 E5 9E5D

3700 3A FB 62 A7 C8 21 17 37 : 75
3708 D5 16 00 5F 19 7E D1 32 : E4
3710 15 62 AF 32 FB 62 C9 53 : D1
3718 55 4D 53 53 41 53 45 41 : 62
3720 53 53 53 49 41 56 52 49 : 74
3728 F5 0F 0F 0F 0F CD 31 37 : 66
3730 F1 E6 0F C6 30 FE 3A 38 : 4C
3738 02 C6 07 CD 54 36 C9 F5 : E4
3740 7C CD 28 37 7D CD 28 37 : 51
3748 F1 C9 C5 3A 45 62 06 27 : 8D
3750 FE 02 28 08 06 22 FE 03 : 59
3758 28 02 06 21 78 CD 54 36 : 20
3760 C1 C9 F5 3E 20 CD 54 36 : 34
3768 F1 C9 E5 21 2E 62 6E 2D : EB
3770 E1 C0 F5 C5 3E 2A CD 54 : E4
3778 36 3E 2A CD 54 36 CD 62 : 24
-----------------------------------
SUM: 10 F8 F0 01 11 58 58 5A A2D4

3780 37 10 F1 C1 F1 C9 3A 13 : 00
3788 62 FE 08 DC D0 37 FE 10 : 59
3790 DC D0 37 3A 9F 60 A7 28 : EB
3798 23 3A 55 60 A7 28 1D 3A : 38
37A0 13 62 47 3E 16 90 47 C5 : AC
37A8 3E 20 CD C8 33 C1 10 F7 : EE
37B0 3E 43 CD C8 33 3E 20 CD : 74
37B8 C8 33 18 08 3A 13 62 FE : C8
37C0 18 DC D0 37 11 05 60 1A : 8B
37C8 13 A7 C8 CD C8 33 18 F7 : 59
37D0 F5 3E 09 CD C8 33 3A 13 : 51
37D8 62 C6 08 E6 F8 32 13 62 : B5
37E0 F1 C9 ED 73 2F 62 3A 9F : 84
37E8 60 A7 C4 05 36 CD 5C 3A : 69
37F0 22 39 62 2A 33 62 23 22 : C1
37F8 33 62 11 05 60 ED 53 31 : 7C
-----------------------------------
SUM: 17 A2 4B 6B 4E 45 A6 BE F689

3800 62 CD 5D 38 CD 53 43 1A : 41
3808 FE 3B 28 3D FE 0D 28 39 : 0A
3810 FE 2E 28 44 D6 41 DA 95 : 1E
3818 48 FE 1A D2 95 48 87 4F : E5
3820 06 00 21 5E 4A 09 7E 23 : 79
3828 66 6F EB CD 77 3A EB DA : 03
3830 95 48 CD 53 43 08 CD 9E : B3
3838 3A CD 53 43 1A A7 28 0F : 95
3840 FE 0D 28 0B FE 3B C2 8A : C3
3848 48 1A 13 FE 0D 20 FA 3A : D4
3850 9F 60 A7 C4 89 36 B7 C9 : A9
3858 21 B5 4B 18 CD 1A FE 40 : 5E
3860 D8 CD 2E 3A CD 53 43 21 : 91
3868 81 38 CD 1F 3A 28 19 21 : 41
3870 84 38 CD 1F 3A 28 63 1A : 87
3878 13 FE 3A CA 0D 39 C3 8A : A8
-----------------------------------
SUM: D7 2F 22 73 03 62 1D 94 CE2B

3880 48 3A 3A 00 45 51 55 00 : A7
```

THE SENTINEL

```
3888 3A 2E 62 A7 28 33 3E 87 : 91
3890 CD DC 34 ED 5B 31 62 CD : 85
3898 B8 39 DA 8C 48 3E 81 CD : 2B
38A0 15 35 ED 5B 31 62 CD 12 : 04
38A8 3A 78 FE 20 D2 92 48 CD : 49
38B0 15 35 1A CD 36 3A 38 06 : DF
38B8 13 CD DC 34 18 F4 13 13 : 22
38C0 C9 CD 5C 3A 44 4D CD 51 : DB
38C8 3A ED 5B 31 62 2A 3B 62 : DC
38D0 23 22 3B 62 CD 52 39 13 : 4D
38D8 13 C9 3A 2E 62 A7 20 17 : 84
38E0 CD FC 44 3A A5 62 E6 01 : 35
38E8 C2 B6 48 D5 3E 01 ED 5B : 1C
38F0 31 62 CD 52 39 D1 C9 3A : BF
38F8 9F 60 A7 28 09 CD FC 44 : E4
---------------------------------
SUM: 16 45 B7 20 5B 86 CF D0 44F1

3900 60 69 CD 1B 36 C9 1A FE : C8
3908 0D C8 13 18 F9 3A 2E 62 : C3
3910 A7 20 29 CD 5C 3A 44 4D : E4
3918 3A 41 62 A7 20 13 ED 5B : FF
3920 31 62 CD 51 3A 2A 3B 62 : B2
3928 23 22 3B 62 CD 52 39 13 : 4D
3930 C9 3E 01 ED 5B 31 62 CD : B0
3938 52 39 13 C9 ED 5B 31 62 : 42
3940 CD B8 39 13 FE 01 C8 3E : D6
3948 87 CD DC 34 3E 81 CD 15 : 05
3950 35 C9 32 B3 39 ED 43 B6 : 02
3958 39 22 B4 39 D5 CD B8 39 : DB
3960 D1 D2 8F 48 D5 CD 40 3A : 96
3968 D1 26 00 6F 29 44 4D 60 : 80
3970 69 CD 58 49 78 B1 20 F7 : 17
3978 2B 2B ED 4B 35 62 CD 46 : 38
---------------------------------
SUM: B5 ED 56 8E EF B8 8A C5 CF00

3980 49 2A 35 62 01 00 00 CD : D8
3988 46 49 3A B3 39 CD 41 49 : 0C
3990 ED 4B B4 39 CD 46 49 ED : 6E
3998 4B B6 39 CD 46 49 1A CD : 7D
39A0 36 3A 38 06 13 CD 41 49 : 18
39A8 18 F4 AF CD 41 49 22 35 : 69
39B0 62 B7 C9 00 00 00 00 00 : E2
39B8 ED 53 10 3A CD 40 3A 26 : F7
39C0 00 6F 29 22 0E 3A 2A 0E : 3A
39C8 3A CD 58 49 78 B1 20 0A : FB
39D0 ED 5B 10 3A CD 2E 3A AF : 76
39D8 37 C9 ED 43 0E 3A 21 07 : A0
39E0 00 09 ED 5B 10 3A 1A CD : 82
39E8 36 3A 38 0A 13 47 CD 53 : 2C
39F0 49 B8 28 F2 18 D0 CD 94 : 64
39F8 1F A7 20 CA 2A 0E 3A 23 : 45
---------------------------------
SUM: 60 AE 07 31 34 64 D4 19 4E36

3A00 23 CD 53 49 CD 58 49 C5 : BF
3A08 CD 58 49 E1 B7 C9 00 00 : CF
3A10 00 00 D5 06 FF 1A 13 04 : 0B
3A18 CD 36 3A 30 F8 D1 C9 D5 : D4
3A20 7E A7 28 08 1A 13 BE 23 : 63
3A28 28 F6 D1 C9 E1 C9 1A CD : 49
3A30 36 3A D8 13 18 F8 FE 40 : A9
3A38 D0 FE 3A 3F D8 FE 30 C9 : 16
3A40 C5 06 00 1A CD 36 3A 38 : 5A
3A48 05 13 80 47 18 F5 78 C1 : 25
3A50 C9 3A 41 62 A7 3E 01 C0 : 4C
3A58 3A 45 62 C9 2A 4C 62 3A : BC
3A60 41 62 A7 C0 2A 46 62 3A : 16
3A68 45 62 FE 02 C8 2A 48 62 : 43
3A70 FE 03 C8 2A 4A 62 C9 23 : 8B
3A78 4D 44 1A B7 37 C8 1A 13 : 8E
---------------------------------
SUM: 07 D3 60 B2 8F 2D CD 5C 5988

3A80 B7 FA 8B 3A BE 20 0C 23 : 83
3A88 C3 7E 3A 08 7E CD 76 43 : 87
3A90 20 07 C9 1A 13 B7 F2 93 : 59
3A98 3A 69 60 C3 7A 3A E6 7F : DF
3AA0 FE 10 38 11 D6 30 38 3F : D4
3AA8 01 02 3B 6F 26 00 29 09 : 05
3AB0 4E 23 66 69 E9 4F 06 00 : 7E
3AB8 21 DA 4B 09 7E CD 7F 34 : 4D
3AC0 C9 3E CB C3 7F 34 3E ED : 73
3AC8 C3 7F 34 79 CD 7F 34 78 : E7
3AD0 C3 7F 34 79 C3 7F 34 FE : 63
3AD8 0F 18 02 FE 0D 3E DD CA : 19
3AE0 7F 34 3E FD C3 7F 34 C6 : 2A
3AE8 20 87 4F 06 00 21 E7 4B : 4F
3AF0 09 7E 23 CD 7F 34 7E CD : 75
3AF8 7F 34 C9 CD 7F 34 CD DA : A3
---------------------------------
SUM: C7 B8 C0 61 09 A2 29 D9 04B6

3B00 35 C9 A2 3D 61 3D 0E 3E : C7
3B08 B6 3F 70 3D FA 3E 2D 3E : 45
3B10 81 3E 6A 42 84 42 2B 43 : 9F
3B18 A9 42 51 3D 8A 48 57 3E : E0
3B20 28 3F 3D 40 85 3E E8 3E : CD
3B28 D6 3B 6D 3D 79 3F BD 40 : 70
3B30 8A 48 D5 3D D2 3D 15 3F : 47
3B38 E3 3F E0 3F DD 3F DA 3F : 76
3B40 B0 3F 13 40 FF 3D 67 3D : 22
3B48 B3 3F D4 3F D1 3F D7 3F : 2B
3B50 64 3D 57 40 5E FE 41 : 15
3B58 22 42 46 42 0A 41 31 41 : A9
3B60 46 41 A2 41 F1 32 E9 41 : B7
3B68 63 40 73 40 B2 40 B8 40 : 40
3B70 4A 3C 4A 3C 72 3C 2C 3C : 22
3B78 98 48 EC 3C EC 3C EC 3C : 58
---------------------------------
SUM: F4 8B FB EC 4F E5 77 F0 E4FC

3B80 EC 3C EC 3C EC 3C EC 3C : A0
3B88 94 3C 6F 3C 6F 3C 30 3D : 93
3B90 30 3D 3D 3E 3D 3E 3D 3E : DE
3B98 3D 3E 98 48 36 3E 36 3E : 43
3BA0 36 3E 36 3E 36 3E 36 3E : D0
3BA8 36 3E 36 3E 42 3E 42 3E : E8
3BB0 4A 3E 4A 3E 67 3E 67 3E : 5A
3BB8 67 3E 67 3E 98 48 60 3E : C8
3BC0 60 3E 60 3E 60 3E 60 3E : 78
3BC8 60 3E 60 3E 60 3E 6C 3E : 84
3BD0 6C 3E 74 3E 74 3E CD 96 : 71
3BD8 43 38 0B FE 0F DC 32 44 : E5
3BE0 01 70 3B C3 AB 3A 21 57 : CC
3BE8 4C CD C4 43 D0 CD 2A 44 : 2B
3BF0 C5 CD 32 44 CD 8F 43 FE : A5
3BF8 0C CA 13 3C FE 04 38 19 : 78
---------------------------------
SUM: 97 B1 D0 34 CE 26 5F F5 9D53

3C00 FE 0D DA 98 48 FE 0F D2 : A4
3C08 98 48 CD DB 3A C1 3E 22 : E3
3C10 C3 FB 3A 3E 32 C1 C3 FB : E7
3C18 3A FE 02 28 F0 87 87 87 : E7
3C20 87 F6 43 F5 CD C6 3A F1 : 73
3C28 C1 C3 FB 3A CD 96 43 38 : 97
3C30 17 0E F9 FE 02 CA D3 3A : F5
3C38 FE 0D 01 DD F9 CA CB 3A : B1
3C40 FE 0E 01 FD F9 CA CB 3A : D2
3C48 3E 03 87 87 87 87 F5 CD : 1F
3C50 3C 44 28 0C F1 F6 01 CD : 69
3C58 7F 34 CD FC 44 C3 DA 35 : 92
3C60 CD C6 3A F1 F6 4B CD 7F : 4B
3C68 34 CD 1F 44 C3 DA 35 CD : 03
3C70 DB 3A CD 3C 44 28 0B 3E : D3
3C78 21 CD 7F 34 CD FC 44 C3 : 71
---------------------------------
SUM: E4 45 3D 14 B8 4A 9E 69 39AA

3C80 DA 35 3E 2A CD 7F 34 CD : C4
3C88 1F 44 C3 DA 35 D6 05 F6 : 06
3C90 78 C3 7F 34 CD 96 43 38 : CC
3C98 12 FE 05 38 5C FE 0D 38 : EC
3CA0 EC FE 0F 38 54 06 38 C5 : 88
3CA8 C3 1F 3D CD 3C 44 28 2A : BE
3CB0 FE 49 28 1A FE 52 20 0B : 04
3CB8 13 CD 75 43 01 ED 5F CA : AF
3CC0 CB 3A 1B CD FC 44 3E 3E : A9
3CC8 CD 7F 34 C3 F3 35 13 CD : 4B
3CD0 75 43 20 EE 01 ED 57 C3 : CE
3CD8 CB 3A 21 76 4C CD C4 43 : BC
3CE0 D0 CD 1F 44 3E 3A CD 7F : C4
3CE8 34 C3 DA 35 D6 05 87 87 : EF
3CF0 87 F5 CD 96 43 38 17 FE : 6F
3CF8 05 DA 98 48 FE 0D 30 1A : 14
---------------------------------
SUM: AB 02 5C 1D 4B 29 6F 26 346A

3D00 D6 05 C1 B0 F6 40 FE 76 : F6
3D08 C2 7F 34 C3 98 48 CD FC : E1
3D10 44 F1 F6 06 CD 7F 34 C3 : 74
3D18 F3 35 FE 0F DA 98 48 CD : BC
3D20 06 44 F1 FE 30 CA 98 48 : 13
3D28 F6 46 CD 7F 34 C3 F3 35 : A7
3D30 CD 06 44 CD 32 44 CD 9D : C4
3D38 43 38 08 F6 70 CD 7F 34 : 69
3D40 C3 F3 35 3E 36 CD 7F 34 : DF
3D48 CD F3 35 CD FC 44 C3 F3 : B8
3D50 35 21 83 4C CD C4 43 D0 : C9
3D58 C3 98 48 0E 88 21 0E 98 : 00
3D60 21 0E A0 21 0E A8 21 0E : D5
3D68 90 21 0E 80 21 0E B0 21 : 3F
3D70 0E B8 CD 96 43 38 10 FE : B2
3D78 0F 30 18 D6 05 38 D9 FE : 41
---------------------------------
SUM: 31 28 BB 3A 39 59 6B 0A CC8B

3D80 08 30 D5 B1 C3 7F 34 79 : AD
3D88 F6 C6 CD 7F 34 CD FC 44 : 49
3D90 C3 F3 35 CD D7 3A 3E 86 : 8D
3D98 B1 CD 7F 34 CD 09 44 C3 : 0E
3DA0 D3 3A CD 8F 43 CD 32 44 : EF
3DA8 FE 0C 28 BE FE 02 28 17 : 2F
3DB0 FE 0D 38 A4 FE 0F 30 A0 : C4
3DB8 F5 CD DB 3A CD 8F 43 C1 : 37
3DC0 B8 20 07 3E 02 18 07 CD : 0B
3DC8 8F 43 FE 04 30 8A 0E 09 : A5
3DD0 18 25 0E C5 21 0E C1 CD : CD
3DD8 8F 43 FE 03 CA 98 48 38 : B5
3DE0 16 FE 04 28 11 FE 0D DA : 36
3DE8 98 48 FE 0F D2 98 48 CD : 6C
3DF0 DB 3A 3E 02 18 01 3D 87 : 32
3DF8 87 87 87 B1 C3 7F 34 CD : 89
---------------------------------
SUM: 34 A8 36 50 82 5A 63 98 0AF6

3E00 8F 43 CD 32 44 FE 0C CA : E9
3E08 5E 3D 0E 42 18 0D CD 8F : 6C
3E10 43 CD 32 44 FE 0C CA 5B : B5
3E18 3D 0E 4A FE 02 C2 98 48 : 37
3E20 CD C6 3A CD 8F 43 FE 04 : 6E
3E28 38 CD C3 98 48 CD 8F 43 : 47
3E30 01 92 3B C3 AB 3A D6 05 : 51
3E38 0E 05 C3 F8 3D 0E 0B C3 : E7
3E40 F7 3D CD DB 3A 3E 2B C3 : 42
3E48 7F 34 CD 06 44 3E 35 CD : 0A
3E50 7F 34 3E 84 C3 DC 34 CD : 15
3E58 8F 43 01 B4 3B C3 AB 3A : 6A
3E60 D6 05 0E 04 C3 F8 3D 0E : F3
3E68 03 C3 F7 3D CD DB 3A 3E : 1A
3E70 23 C3 7F 34 CD 06 44 3E : EE
3E78 34 CD 7F 34 3E 84 C3 DC : 15
---------------------------------
SUM: 35 C5 2E 98 32 A9 66 08 49F4

3E80 34 0E 10 18 15 CD 88 43 : 17
3E88 0E 18 38 0E FE 04 D2 98 : D8
3E90 48 CD 32 44 87 87 87 F6 : 16
3E98 20 4F 79 CD 7F 34 CD 94 : C9
3EA0 47 3A 2E 62 A7 CA 7F 34 : 35
3EA8 3A 41 62 A7 28 0A 3A 4E : 3E
3EB0 62 FE 01 C2 B9 48 18 0B : 47
3EB8 3A 4E 62 47 3A 45 62 B8 : CA
3EC0 C2 B9 48 E5 CD 5C 3A 44 : 4F
3EC8 4D E1 37 ED 42 38 0D 25 : FE
3ED0 24 C2 A1 48 7D B7 F2 7F : 74
3ED8 34 C3 A1 48 24 C2 A1 48 : AF
3EE0 7D B7 FA 7F 34 C3 A1 48 : 8D
3EE8 CD 88 43 0E C2 30 12 21 : CB
3EF0 B3 4C CD C4 43 D0 3E C3 : A4
3EF8 18 11 CD 88 43 38 0A 0E : 11
---------------------------------
SUM: 43 C4 7E 84 07 F5 B6 14 0482

3F00 C4 CD F8 3D CD 32 44 18 : 21
3F08 05 3E CD CD 7F 34 CD FC : 59
3F10 44 CD DA 35 C9 CD 75 43 : 6E
3F18 3E C9 CA 7F 34 CD 88 43 : 1C
3F20 DA 98 48 0E C0 C3 F8 3D : 80
3F28 CD 96 43 DA 98 48 CD 32 : 5F
3F30 44 D6 05 DA 98 48 FE 06 : DD
3F38 CA 98 48 FE 07 28 14 D2 : BD
3F40 98 48 47 CD C6 3A 78 0E : 7A
3F48 40 CD F8 3D CD 65 3F C8 : 7B
3F50 C3 98 48 21 72 3F CD C4 : 06
3F58 43 D0 3E DB CD 7F 34 CD : 79
3F60 2A 44 C3 F3 35 21 72 3F : 2B
3F68 06 03 1A BE C0 13 23 10 : E7
3F70 F9 C9 28 43 29 00 ED 78 : BB
3F78 00 D5 CD 65 3F 28 1A D1 : 59
---------------------------------
SUM: 07 9F D8 DD 6F 34 39 E0 03B3

3F80 3E D3 CD 7F 34 CD 2A 44 : CC
3F88 CD 32 44 CD 9D 43 DA 98 : 62
3F90 48 FE 07 C2 98 48 C3 F3 : A5
3F98 35 C1 CD C6 3A CD 32 44 : 06
3FA0 CD 9D 43 DA 98 48 FE 06 : 6B
3FA8 0E 41 C2 F8 3D C3 98 48 : E9
3FB0 0E 80 21 0E C0 21 0E 40 : EC
3FB8 1A CD 39 3A DA 98 48 FE : 12
3FC0 38 D2 98 48 D6 30 87 87 : FE
3FC8 87 B1 4F 13 CD 32 44 18 : F5
3FD0 14 0E 28 21 0E 20 21 0E : C8
3FD8 38 21 0E 08 21 0E 00 21 : BF
3FE0 0E 18 21 0E 10 CD 8F 43 : 04
3FE8 FE 0F 30 13 D6 05 DA 98 : 9D
3FF0 48 FE 08 D2 98 48 47 CD : 14
3FF8 C1 3A 78 B1 C3 7F 34 C5 : 5F
---------------------------------
SUM: AB 00 32 16 25 12 B5 DA 8E90

4000 CD D7 3A CD C1 3A CD 09 : 7C
4008 44 CD D3 3A C1 79 F6 06 : 54
4010 C3 7F 34 AF 32 A5 62 CD : 2B
4018 53 43 CD E7 46 24 25 C2 : 9B
4020 98 48 7D FE 08 38 0E 6F : 18
4028 E6 F8 BD C2 98 48 FE 40 : 7B
4030 38 06 C3 98 48 87 87 87 : 76
4038 F6 C7 C3 7F 34 1A 13 D6 : 36
4040 30 28 0C DA 98 48 FE 03 : 1F
4048 D2 98 48 3C 87 87 87 F6 : 79
4050 46 47 0E ED C3 CB 3A 3A : 8A
4058 9F 60 32 3F 62 C9 AF 32 : 7C
4060 3F 62 C9 CD FC 44 3A A5 : 56
4068 62 E6 01 C2 B6 48 ED 43 : 39
4070 A6 62 C9 CD 53 43 06 50 : 8A
4078 21 AA 62 1A 13 FE 27 28 : A7
---------------------------------
SUM: 22 2E 57 2C 72 CD B2 6F 828E

4080 10 FE 22 28 1C 77 23 1A : 28
4088 13 FE 0D 28 22 10 F6 18 : 86
4090 1E 1A 13 FE 27 28 18 FE : AE
4098 0D 28 14 77 23 10 F2 18 : FD
40A0 0E 1A 13 FE 22 28 08 FE : 89
40A8 0D 28 04 77 23 10 F2 36 : 0B
40B0 00 C9 3E 01 32 40 62 C9 : A5
40B8 AF 32 40 62 C9 3A 9F 60 : 85
40C0 A7 C4 36 36 CD FC 44 CD : B1
40C8 C8 35 3A A5 62 E6 01 C2 : E7
40D0 B6 48 ED 43 4C 62 3A 41 : 57
40D8 62 F6 01 32 41 62 6F 3E : DB
40E0 E8 CD DC 34 3A 45 62 FE : A4
40E8 02 20 09 ED 43 46 62 7D : 80
40F0 32 42 62 C9 FE 03 20 09 : C9
40F8 ED 43 48 62 7D 32 43 62 : 2E
---------------------------------
SUM: A8 24 D8 39 7C D7 33 99 220F

4100 C9 ED 43 4A 62 7D 32 43 : 97
4108 62 C9 3A 9F 60 A7 C4 36 : 05
4110 36 CD FC 44 CD C8 35 3A : 47
4118 A5 62 E6 01 C2 B6 48 ED : 9B
4120 43 4C 62 3A 41 62 F6 02 : C6
4128 32 41 62 3E E2 CD DC 34 : D2
4130 C9 3A 9F 60 A7 C4 36 36 : D9
4138 3A 41 62 E6 FD 32 41 62 : 95
4140 3E E3 CD DC 34 C9 3A 2E : 2F
4148 62 A7 20 15 CD 53 43 3E : DF
4150 05 2A 3B 62 23 22 3B 62 : AE
4158 CD 52 39 CD 5D 43 28 EC : D9
4160 C9 3A 9F 60 A7 C4 36 36 : D9
4168 CD B8 39 A7 C4 75 41 CD : AC
4170 5D 43 28 ED C9 ED 5B 10 : D6
```

▶泉大介氏のColumnsを解凍したとたん，200年の眠りから覚めた悪霊が妻に乗り移った。昼間は普段と変わりないのだが，夜まわりが寝静まるとコンソールに問いだす。“ピッ”“ピッ”“ピワァ～ン”。「おい，寝ないのか？」すると妻は振り向きもせずキーボードを叩き続けながら言った。「YET！」

長谷川　英治（36）大阪府

```
4178 3A CD 12 3A 78 FE 20 D2 : BB
---------------------------------
SUM: 1D F5 97 3A 45 6C 8E 0D D44D

4180 92 48 F6 20 CD DC 34 1A : E7
4188 CD 36 3A 38 06 13 CD DC : 37
4190 34 18 F4 3E 81 CD 15 35 : 16
4198 2A 0E 3A 23 23 AF CD 9A : CE
41A0 1F C9 3A 2E 62 A7 CA 49 : 6C
41A8 38 CD 53 43 D5 CD B8 39 : 2E
41B0 DA 8C 48 D1 FE 01 20 13 : B1
41B8 60 69 3E E7 CD 15 35 2A : 2F
41C0 3B 62 23 22 3B 62 3E 81 : 3E
41C8 CD 15 35 CD 12 3A 78 FE : A6
41D0 20 D2 92 48 CD 15 35 1A : FD
41D8 CD 36 3A 38 06 13 CD DC : 37
41E0 34 18 F4 CD 5D 43 28 C1 : 96
41E8 C9 3E 01 32 4F 62 CD 42 : FA
41F0 44 C8 CD FC 44 CD C8 35 : E3
41F8 3E E9 CD DC 34 C9 3A 41 : 48
---------------------------------
SUM: C2 B5 24 28 BD F4 69 72 68D2

4200 62 E6 02 C2 B9 48 3E 02 : 4D
4208 32 45 62 3A 42 62 32 41 : 2A
4210 62 2A 46 62 22 4C 62 3E : 42
4218 E1 CD DC 34 3E 00 CD DC : A5
4220 34 C9 3A 41 62 E6 02 C2 : 84
4228 B9 48 3E 03 32 45 62 3A : 55
4230 43 62 32 41 62 2A 48 62 : 4E
4238 22 4C 62 3E E1 CD DC 34 : CC
4240 3E 01 CD DC 34 C9 3A 41 : 60
4248 62 E6 02 C2 B9 48 3E 04 : 4F
4250 32 45 62 3A 44 62 32 41 : 2C
4258 62 2A 4A 62 22 4C 62 3E : 46
4260 E1 CD DC 34 3E 02 CD DC : A7
4268 34 C9 CD 53 43 1A CD 82 : C9
4270 43 20 05 CD B0 42 18 06 : 45
4278 CD FC 44 CD F3 35 CD 5D : 2C
---------------------------------
SUM: 82 E9 FF B0 A9 6A B2 74 3241

4280 43 28 E7 C9 CD 53 43 CD : 4B
4288 FC 44 3A A5 62 E6 01 20 : 88
4290 0A 79 CD 7F 34 78 CD 7F : C7
4298 34 18 08 CD C8 35 3E E5 : 41
42A0 CD DC 34 CD 5D 43 28 DC : 4E
42A8 C9 1A CD 82 43 C2 AD 48 : 2C
42B0 3A 40 62 A7 20 28 13 1A : F8
42B8 FE 0D CA AD 48 FE 5E 28 : 4E
42C0 18 CD 82 43 20 0E 13 4F : 3A
42C8 1A CD 82 43 28 04 CD 6A : 0F
42D0 43 C8 1B 79 CD 7F 34 18 : 37
42D8 DD CD 1A 43 18 F6 13 1A : 42
42E0 FE 0D CA AD 48 FE 5E 28 : 4E
42E8 2C CD 44 43 38 0F CD 7F : 13
42F0 34 13 1A FE 0D CA AD 48 : 2B
42F8 CD 7F 34 18 E1 CD 82 43 : 0B
---------------------------------
SUM: C8 DB B8 A5 CE 3C 16 D4 088B

4300 20 0E 13 4F 1A CD 82 43 : 3C
4308 28 04 CD 6A 43 C8 1B 79 : 02
4310 CD 7F 34 18 C9 CD 1A 43 : 8B
4318 18 F6 13 1A FE 0D CA AD : BD
4320 48 D6 40 38 03 FE 1C D8 : 8B
4328 C6 40 C9 CD FC 44 CD C8 : 71
4330 35 3E B0 CD DC 34 3A A5 : DF
4338 62 E6 01 C2 B6 48 60 69 : D2
4340 CD BC 44 C9 FE 81 D8 FE : EB
4348 A0 3F D0 FE E0 D8 FE F0 : 53
4350 3F C9 13 1A FE 09 28 FA : 5E
4358 FE 20 28 F6 C9 CD 53 43 : 68
4360 FE 2C 28 03 FE 3A C0 13 : 60
4368 C9 1A D9 21 C9 4C 01 0E : 01
4370 00 ED B1 D9 C9 1A FE 09 : 61
4378 C8 FE 0D C8 FE 3B C8 FE : 9A
---------------------------------
SUM: 0B D6 EF 1B E8 37 DC AD E462

4380 20 C9 FE 27 C8 FE 22 C9 : BF
4388 21 42 4C 06 08 18 13 CD : B5
4390 96 43 D0 C3 98 48 21 11 : 7E
4398 4C 06 11 18 05 21 20 4C : 0D
43A0 06 08 C5 48 D5 1A BE 20 : E8
43A8 04 13 23 18 F8 34 35 20 : D3
43B0 0A CD 6A 43 20 05 E1 79 : 03
43B8 90 C1 C9 CD F8 43 D1 10 : 03
43C0 E3 C1 37 C9 34 35 37 C8 : 0C
43C8 4B 42 1A BE 20 05 13 23 : C0
43D0 C3 CA 43 34 35 23 20 05 : 81
43D8 CD 76 43 28 0A CD F8 43 : C0
43E0 23 23 59 50 C3 C4 43 4E : 07
43E8 23 46 78 B7 20 05 CD D3 : 5D
43F0 3A B7 C9 CD CB 3A B7 C9 : 0C
43F8 AF BE 23 C8 C3 F9 43 CD : 24
---------------------------------
SUM: B4 1E DA F7 56 3B 87 A6 AA6F

4400 1F 44 AF 91 4F C9 CD D7 : 5F
4408 3A 1A 13 FE 2B 28 10 FE : C6
4410 2D 28 EC FE 29 C2 98 48 : 0A
4418 AF 32 A5 62 0E 00 C9 CD : 8C
4420 FC 44 1A FE 29 C2 A7 48 : 32
4428 13 C9 CD 3C 44 C2 98 48 : CB
4430 18 ED 47 1A FE 2C 78 C2 : CA
4438 AA 48 13 C9 1A FE 28 C0 : CE
4440 13 C9 CD 53 43 FE 0D C8 : 12
4448 FE 3B C9 21 00 00 22 3B : 80
4450 62 22 A8 62 22 3D 62 21 : 70
4458 00 02 22 35 62 21 FF FF : DA
4460 22 A6 62 AF 32 AA 62 AF : C6
4468 32 50 62 32 52 62 32 42 : 3E
4470 62 32 43 62 32 44 62 32 : 43
4478 41 62 3E 02 32 45 62 3E : FA
---------------------------------
SUM: 70 AC 39 5C E5 52 05 80 EBC4

4480 00 32 40 62 21 00 00 22 : 17
4488 33 62 22 46 62 22 48 62 : 2B
4490 22 4A 62 22 4C 62 C9 F5 : 5C
4498 E5 3A 45 62 21 46 62 FE : 8D
44A0 02 28 0A 21 48 62 FE 03 : 00
44A8 28 03 21 4A 62 34 20 02 : 4E
44B0 23 34 2A 4C 62 23 22 4C : C0
44B8 62 E1 F1 C9 E5 ED 4B 4C : 66
44C0 62 09 22 4C 62 E1 3A 45 : 9B
44C8 62 FE 02 20 09 ED 4B 46 : 09
44D0 62 09 22 46 62 C9 FE 03 : FF
44D8 20 09 ED 4B 48 62 09 22 : 36
44E0 48 62 C9 ED 4B 4A 62 09 : 60
44E8 22 4A 62 C9 21 00 00 01 : B9
44F0 01 02 AF CD 9A 1F ED A1 : C6
44F8 EA F3 44 C9 AF 32 51 62 : 7E
---------------------------------
SUM: 84 12 A0 F5 AB 04 2A D1 7D41

4500 32 A5 62 CD 2B 45 CD 53 : 96
4508 43 FE 2B 20 0A E5 CD 2A : 72
4510 45 C1 CD 33 47 18 EF FE : 52
4518 2D 20 0C E5 CD 2A 45 4D : C7
4520 44 E1 CD 3A 47 18 DF 4D : B7
4528 44 C9 13 CD 84 45 CD 53 : D6
4530 43 FE 2A 20 0A E5 CD 83 : CA
4538 45 C1 CD 43 47 18 EF FE : 62
4540 2F 20 0C E5 CD 83 45 4D : 22
4548 44 E1 CD 51 47 18 DF FE : 7F
4550 25 C0 E5 CD 83 45 4D 44 : F0
4558 E1 D5 CD 5F 47 D1 18 CE : E0
4560 CD 83 45 7C 2F 67 7D 2F : 53
4568 6F 23 C9 4F 13 1A FE 0D : E2
4570 CA AD 48 FE 5E CC 1A 43 : 44
4578 26 00 6F 13 1A 13 B9 C2 : 50
---------------------------------
SUM: 9C D6 8D AD FD D7 0D 87 D992

4580 AD 48 C9 13 CD 53 43 FE : 32
4588 28 CA 47 46 FE 2D CA 6D : E1
4590 47 FE 24 CA 80 47 FE 25 : 1D
4598 CA 7B 47 CD 39 3A D2 86 : 24
45A0 47 CD 82 43 CA 8B 47 21 : 96
45A8 CB 4C 01 0C 00 ED B1 CA : 8C
45B0 A4 48 C5 3A 2E 62 A7 20 : 42
45B8 29 3A A5 62 F6 02 32 A5 : 39
45C0 62 CD 59 46 28 10 CD B8 : 8B
45C8 39 38 04 FE 01 28 39 3E : 13
45D0 03 32 A5 62 18 32 3E 03 : C7
45D8 32 A5 62 CD 2E 3A 13 13 : 94
45E0 18 26 3A A5 62 F6 02 32 : A9
45E8 A5 62 CD 59 46 28 26 CD : 8E
45F0 B8 39 DA 8C 48 FE 01 28 : C6
45F8 13 FE 05 CA 38 46 3E 03 : 9F
---------------------------------
SUM: 1D C1 B2 A2 09 E3 6C FC 8C15

4600 32 A5 62 3E 80 CD BA 35 : B3
4608 60 69 C1 C9 60 69 3E E7 : 41
4610 CD BA 35 C1 C9 CD 12 3A : 5F
4618 78 FE 20 D2 92 48 F6 20 : 58
4620 CD 9C 35 1A CD 36 3A 38 : 2D
4628 06 13 CD 9C 35 18 F4 3E : 01
4630 03 32 A5 62 13 13 C1 C9 : EC
4638 3E 03 32 A5 62 3E 80 CD : 05
4640 BA 35 CD 75 41 C1 C9 C5 : C1
4648 13 CD 03 45 60 69 C1 CD : 7F
4650 53 43 13 FE 29 C2 A7 48 : 81
4658 C9 D5 CD 2E 3A 1A 13 FE : FE
4660 23 20 04 1A 13 FE 23 D1 : 66
4668 C9 13 1A CD B8 1F D2 C1 : 2D
4670 46 2A 39 62 3A A5 62 F6 : 42
4678 02 32 A5 62 3A 45 62 47 : 63
---------------------------------
SUM: 08 53 FD E8 F5 F7 6C 29 5152

4680 3A 4E 62 A7 28 04 B8 C2 : 37
4688 A4 48 78 32 4E 62 C9 13 : 22
4690 1A CD B8 1F 30 2B 2A 39 : 7C
4698 62 3A 45 62 FE 01 C8 D5 : DF
46A0 3E 87 CD 9C 35 CD 5C 3A : C6
46A8 ED 5B 39 62 B7 ED 52 3E : 17
46B0 E7 CD BA 35 3E C1 CD 9C : 0B
46B8 35 D1 3E 03 32 A5 62 B7 : 37
46C0 C9 6F 26 00 13 1A CD B8 : 10
46C8 1F D8 29 29 29 29 B5 6F : BF
46D0 13 18 F2 21 00 00 13 1A : 6B
46D8 FE 5F 28 FA D6 30 D8 FE : 5B
46E0 02 D0 0F ED 6A 18 EF D5 : 14
46E8 D6 30 4F 13 CD 69 43 20 : 01
46F0 FA 1B 1A D1 FE 48 20 0C : 72
46F8 79 CD C1 46 1A FE 48 C2 : 6F
---------------------------------
SUM: E5 C3 77 EB 61 EC 57 B0 FB1D

4700 A4 48 13 C9 FE 42 20 14 : 3C
4708 79 FE 02 D2 A4 48 6F 26 : CC
4710 00 CD D6 46 1A FE 42 C2 : 05
4718 A4 48 13 C9 69 26 00 13 : 6A
4720 1A CD 39 3A D8 D6 30 4D : 85
4728 44 29 29 09 29 4F 06 00 : 1D
4730 09 18 EC 09 3E C0 CD 9C : 7D
4738 35 C9 B7 ED 42 3E C1 CD : B0
4740 9C 35 C9 3A A5 62 E6 02 : C3
4748 CC 46 48 3E C2 CD 9C 35 : F8
4750 C9 3A A5 62 E6 02 CC 59 : 17
4758 48 3E C3 CD 9C 35 C9 3A : EA
4760 A5 62 E6 02 CC 70 48 3E : B1
4768 C7 CD 9C 35 C9 3A A5 62 : 6F
4770 E6 02 CC 60 45 3E C6 CD : 2A
4778 9C 35 C9 CD D3 46 18 0E : A6
---------------------------------
SUM: C4 8B 93 EE 3C 65 77 0A 8FA5

4780 CD 8F 46 38 09 C9 CD E7 : 60
4788 46 18 03 CD 6B 45 3E E7 : 03
4790 CD BA 35 C9 AF 32 A5 62 : 6D
4798 32 4E 62 32 51 62 3A 2E : 2F
47A0 62 A7 CA FC 44 CD C7 47 : EE
47A8 CD 53 43 FE 2B 20 08 E5 : 99
47B0 CD C6 47 C1 09 18 F1 FE : AB
47B8 2D C0 E5 CD C6 47 44 4D : 3D
47C0 E1 B7 ED 42 18 E2 13 CD : A1
47C8 53 43 FE 2D CA 60 45 FE : 2E
47D0 24 CA 69 46 FE 25 CA D3 : 5D
47D8 46 CD 39 3A D2 E7 46 CD : 52
47E0 82 43 CA 6B 45 21 CB 4C : 77
47E8 01 0C 00 ED B1 CA A4 48 : 61
47F0 3A A5 62 F6 02 32 A5 62 : 72
47F8 CD B8 39 DA 8C 48 A7 CA : DD
---------------------------------
SUM: 63 6C 0B 9F E8 A1 11 00 03E7

4800 92 48 FE 05 CA 92 48 60 : E1
4808 69 47 3A 4E 62 A7 28 04 : 6D
4810 B8 C2 A4 48 78 32 4E 62 : C0
4818 C9 D5 F5 AF 32 05 63 01 : DD
4820 0A 05 11 05 63 CD 60 48 : FD
4828 C6 30 1B 12 10 F7 6B 62 : F7
4830 01 20 04 3E 30 BE 20 04 : 75
4838 71 23 10 F9 F1 B7 28 01 : 6E
4840 EB CD E5 1F D1 C9 D5 EB : 16
4848 21 00 00 3E 10 29 EB 29 : AC
4850 EB 30 01 09 3D 20 F6 D1 : 49
4858 C9 D5 CD 70 48 EB D1 C9 : A8
4860 C5 AF 06 10 29 17 2C 91 : 87
4868 30 02 2D 81 10 F6 C1 C9 : 70
4870 78 B1 28 30 EB 21 00 00 : 8D
4878 3E 10 EB 29 EB ED 6A 1C : C0
---------------------------------
SUM: 29 E2 0A 58 DF C1 12 9A A358

4880 ED 42 30 02 09 1D 3D 20 : E4
4888 F1 C9 AF 01 3E 01 01 3E : E8
4890 02 01 3E 03 01 3E 04 01 : 88
4898 3E 05 01 3E 06 01 3E 07 : CE
48A0 01 3E 08 01 3E 09 01 3E : CE
48A8 0A 01 3E 0B 01 3E 0C 01 : A0
48B0 3E 0D 01 3E 0E 01 3E 0F : E6
48B8 01 3E 10 32 FB 62 ED 53 : 1E
48C0 FE 62 87 21 65 49 4F 06 : 0B
48C8 00 09 5E 23 56 CD E5 1F : B1
48D0 11 55 4A CD E5 1F CD FC : 4A
48D8 48 3A FD 62 47 CD DF 1F : F3
48E0 3E 5E CD F4 1F CD EE 1F : 56
48E8 2A 3D 62 23 22 3D 62 ED : 9A
48F0 5B 31 62 CD 49 38 ED 7B : A4
48F8 2F 62 B7 C9 D5 CD F1 1F : C3
---------------------------------
SUM: B1 C3 E9 E0 DC 18 C6 ED CB67

4900 2A 33 62 AF CD 19 48 CD : 69
4908 F1 1F CD F1 1F ED 5B 31 : 66
4910 62 2A FE 62 B7 ED 52 20 : 02
4918 07 2A 7A 1F 7E 32 FD 62 : D9
4920 1A 13 FE 0D 28 0A FE 09 : 71
4928 C4 F4 1F CC 35 49 18 E1 : 1A
4930 D1 CD EE 1F C9 CD F1 1F : 51
4938 2A 7A 1F 7E E6 07 20 F5 : 43
4940 C9 CD 9A 1F 23 C9 F5 79 : A9
4948 CD 9A 1F 23 78 CD 9A 1F : A7
4950 23 F1 C9 CD 94 1F 23 C9 : 49
4958 F5 CD 94 1F 4F 23 CD 94 : 48
4960 1F 47 23 F1 C9 87 49 8E : A1
4968 49 9E 49 AB 49 B9 49 C8 : EE
4970 49 D8 49 E8 49 F6 49 FE : D8
4978 49 11 4A 1D 4A 29 4A 37 : B5
---------------------------------
SUM: 05 E7 E6 66 50 83 BD FE 2E3F

4980 4A 38 4A 44 4A 4A 4A 53 : 41
4988 79 6E 74 61 78 00 55 6E : F7
4990 64 65 66 69 6E 65 64 20 : EF
4998 4C 61 62 65 6C 00 52 65 : 97
49A0 64 65 66 69 6E 69 74 69 : 4C
49A8 6F 6E 00 49 6C 6C 65 67 : CA
49B0 61 6C 20 4C 61 62 65 6C : CD
49B8 00 49 6C 6C 65 67 61 6C : BA
49C0 20 4F 70 63 6F 64 65 00 : 7A
49C8 49 6C 6C 65 67 61 6C 20 : DA
49D0 4F 70 65 72 61 6E 64 00 : C9
49D8 54 6F 6F 20 4D 61 6E 79 : E7
49E0 20 4C 61 62 65 6C 73 00 : 73
49E8 4D 69 73 73 69 6E 67 20 : FA
49F0 4C 61 62 65 6C 00 54 6F : A3
49F8 6F 20 46 61 72 00 49 6C : 5D
---------------------------------
SUM: DB C4 A4 D2 6C BB 0E 82 813E

4A00 6C 65 67 61 6C 20 45 78 : E2
4A08 70 72 65 73 73 69 6F 6E : 73
4A10 00 4D 69 73 73 69 6E 67 : DA
4A18 20 5B 29 5D 00 4D 69 73 : 2A
4A20 73 69 6E 67 20 5B 2E 5D : B7
4A28 00 4D 69 73 73 69 6E 67 : DA
4A30 20 51 75 6F 74 65 00 00 : 2E
4A38 49 6C 6C 65 67 61 6C 20 : DA
4A40 4F 52 47 00 56 61 6C 75 : 80
```


▶うちのパソコンはコタツの上に置いているけど、読者の皆さんはどこに置いているのでしょうか。ぜひ調べて発表してください。コタツでパソコンは腰にきます。
奥江　聡（20）神奈川県

```
4A48 65 00 52 65 6C 6F 63 61 : BB
4A50 74 69 6F 6E 00 20 45 72 : 91
4A58 72 6F 72 20 0D 00 8E 4A : 58
4A60 98 4A 9C 4A C1 4A EA 4A : 07
4A68 D9 4B D9 4B FB 4A 00 4B : D8
4A70 1D 4B D9 4B 22 4B D9 4B : 1D
4A78 33 4B 3A 4B 53 4B D9 4B : C5
--------------------------------
SUM: 33 47 18 70 C0 E3 D1 61 D796

4A80 65 4B 95 4B AB 4B D9 4B : AA
4A88 D9 4B D9 4B B1 4B 44 44 : CC
4A90 B0 4E 44 B1 44 43 B2 00 : 2C
4A98 49 54 B3 00 50 B4 41 4C : E1
4AA0 4C B5 43 46 80 50 4C 81 : 27
4AA8 50 49 90 50 44 91 50 49 : E7
4AB0 52 92 50 44 52 93 53 45 : F5
4AB8 47 DA 4F 4D 4D 4F 4E DC : 83
4AC0 00 45 43 B6 4A 4E 5A B7 : E7
4AC8 45 46 42 B8 45 46 57 B9 : 20
4AD0 45 46 53 BA 42 46 57 B9 : A2
4AD8 53 BA 41 41 83 49 82 4D : 2A
4AE0 BB 45 46 4D BB 53 45 47 : 2D
4AE8 DB 00 58 BC 58 58 84 51 : 74
4AF0 55 BD 49 85 58 54 DF 4E : B9
4AF8 44 E2 00 41 4C 54 86 00 : 8D
--------------------------------
SUM: 78 11 D7 A6 5E 38 05 22 1AC3

4B00 4E 43 BE 4E BF 4E 49 94 : 87
4B08 4E 44 95 4E 49 52 96 4E : F4
4B10 44 52 97 4D C0 4E 43 4C : 17
4B18 55 44 45 E1 00 52 C1 50 : 22
4B20 C2 00 44 C3 44 49 52 98 : 40
4B28 44 49 99 44 44 52 9A 44 : DE
4B30 44 9B 00 45 47 9C 4F 50 : A6
4B38 87 00 52 C4 55 54 C5 55 : 60
4B40 54 49 9D 54 49 52 9E 55 : 1C
4B48 54 44 9F 54 44 52 A0 52 : 13
4B50 00 47 C6 00 4F 50 C8 55 53 : 1C
4B58 48 C9 55 42 4C 49 43 E0 : 60
4B60 41 47 45 E3 00 45 54 CA : 13
4B68 4C 41 88 52 41 89 4C 43 : C0
4B70 41 8A 52 43 41 8B 4C CB : 43
4B78 52 CC 4C 43 CD 52 43 CE : DD
--------------------------------
SUM: 5D FB 5A CE 64 2B E8 7F D107

4B80 4C 44 A1 52 44 A2 45 53 : 01
4B88 CF 53 54 D0 45 54 49 A3 : CB
4B90 45 54 4E A4 00 42 43 D1 : E1
4B98 55 42 D2 43 46 8C 45 54 : 17
4BA0 D3 4C 41 D4 52 41 D5 52 : EE
4BA8 4C D6 00 49 54 4C 45 E4 : 34
4BB0 00 4F 52 D7 00 4C 49 53 : 60
4BB8 54 D8 4E 4C 49 53 54 D9 : 8F
4BC0 50 48 41 53 45 DD 44 45 : D7
4BC8 50 48 41 53 45 DE 4B 41 : DB
4BD0 4E 4A 49 4E 4B 41 4E 4A : 53
4BD8 49 00 3F 2F F3 27 D9 FB : A5
4BE0 76 00 17 1F 07 0F 37 ED : E6
4BE8 A1 ED A9 ED B1 ED B9 ED : 68
4BF0 A2 ED AA ED B2 ED BA ED : 6C
4BF8 B0 ED A0 ED B8 ED A8 ED : 64
--------------------------------
SUM: C8 17 0A 52 A8 E9 D5 FC FEED

4C00 44 ED A3 ED B3 ED AB ED : F9
4C08 BB ED 6F ED 67 ED 4D ED : 92
4C10 45 42 43 00 44 45 00 48 : 9B
4C18 4C 00 53 50 00 41 46 00 : 76
4C20 42 00 43 00 44 00 45 00 : 0E
4C28 48 00 4C 00 28 48 4C 29 : 79
4C30 00 41 00 49 58 00 49 59 : 84
4C38 00 28 49 58 00 28 49 59 : 93
4C40 00 00 4E 5A 00 5A 00 4E : 50
4C48 43 00 43 00 50 4F 00 50 : 75
4C50 45 00 50 00 4D 00 00 28 : 0A
4C58 44 45 29 2C 41 00 12 00 : 31
4C60 28 42 43 29 2C 41 00 02 : 45
4C68 00 52 2C 41 00 ED 4F 49 : 44
4C70 2C 41 00 ED 47 00 44 45 : 2A
4C78 29 00 1A 00 42 43 29 00 : F1
--------------------------------
SUM: 63 9F 13 A8 B5 EA 2F 53 FB1C

4C80 0A 00 00 44 45 2C 48 4C : 53
4C88 00 EB 00 41 46 2C 41 46 : 25
4C90 27 00 08 00 28 53 50 29 : 23
4C98 2C 48 4C 00 E3 00 28 53 : 1E
4CA0 50 29 2C 49 58 00 DD E3 : 06
4CA8 28 53 50 29 2C 49 59 00 : C2
4CB0 FD E3 00 28 48 4C 29 00 : C5
4CB8 E9 00 28 49 58 29 00 DD : B8
4CC0 E9 28 49 59 29 00 FD E9 : C2
4CC8 00 09 20 0D 3B 3A 29 2C : 00
4CD0 2B 2D 2A 2F 25 22 27 22 : 41
4CD8 A8 50 CD A3 1F D8 CD 3F : 6B
4CE0 4D D8 2A 74 1F 11 06 63 : 5C
4CE8 01 20 00 ED B0 3A 5D 1F : 74
4CF0 32 37 63 AF 32 AA 50 CD : 74
4CF8 43 50 D8 06 10 0E 00 3A : C9
--------------------------------
SUM: 3A BF BD B6 73 A0 2D CD 5555

4D00 24 63 11 26 63 12 13 FE : 44
4D08 7F 30 0F 2A 62 1F 85 6F : 5D
4D10 30 01 24 7E 05 28 24 0C : 30
4D18 18 EB 0D F5 79 87 87 87 : 13
4D20 87 4F F1 D6 80 81 32 3D : 0D
4D28 63 AF 32 38 63 01 38 00 : 18
4D30 ED 5B A8 50 21 06 63 ED : B7
4D38 B0 A7 C9 3E 07 37 C9 3A : 9F
4D40 5D 1F CD 5E 4D D8 CD 69 : 02
4D48 4D D8 3E 08 37 C0 E5 ED : 34
4D50 5B 74 1F 01 20 00 ED B0 : AC
4D58 E1 7E CD B3 4D C9 FE 41 : 34
4D60 38 04 FE 45 3F D0 3E 03 : CF
4D68 C9 0E 10 ED 5B 60 1F ED : 9B
4D70 53 39 63 2A 64 1F 3E 01 : DB
4D78 CD 00 20 D8 06 08 22 3B : 30
--------------------------------
SUM: 79 B3 6D AD 43 57 33 D7 1818

4D80 63 7E FE FF 28 1A B7 28 : FF
4D88 0B D5 ED 5B 74 1F CD A4 : 2C
4D90 4D D1 28 0D D5 11 20 00 : 59
4D98 19 D1 10 E2 13 0D 20 CF : EB
4DA0 3E AF B7 C9 C5 E5 06 10 : 2D
4DA8 13 23 1A BE 20 02 10 F8 : 38
4DB0 E1 C1 C9 E5 E6 87 21 1F : FD
4DB8 29 BE E1 C8 3E 06 37 C9 : D4
4DC0 22 A8 50 CD A3 1F D8 CD : 4E
4DC8 AF 1F D8 01 20 00 11 06 : DE
4DD0 63 2A 74 1F ED B0 3A 5D : 54
4DD8 1F 32 37 63 2A E1 27 22 : 3F
4DE0 3B 63 2A DF 27 22 39 63 : 8C
4DE8 CD 72 50 D8 16 00 5F 2A : 06
4DF0 62 1F 19 36 8F 32 24 63 : 18
4DF8 32 26 63 3E 80 32 27 63 : 35
--------------------------------
SUM: 1E 83 67 F8 B3 01 5F 30 2FB1

4E00 CD 61 50 D8 AF 32 3D 63 : D7
4E08 32 AA 50 32 38 63 21 00 : 1A
4E10 00 22 18 63 01 20 00 ED : AB
4E18 5B 3B 63 2A 74 1F ED B0 : 53
4E20 3E 01 ED 5B 39 63 2A 64 : B1
4E28 1F CD 03 20 D8 01 38 00 : 20
4E30 ED 5B A8 50 21 06 63 ED : B7
4E38 B0 B7 C9 22 A8 50 11 06 : 61
4E40 63 01 38 00 ED B0 3A 37 : AA
4E48 63 32 5D 1F 2A 18 63 2C : E2
4E50 2D C4 87 4F 3A 3D 63 2A : CB
4E58 18 63 2C 2D 20 01 3C 67 : 98
4E60 22 18 63 3E 01 ED 5B 39 : 5D
4E68 63 2A 64 1F CD 20 00 D8 : D5
4E70 21 06 63 ED 5B 3B 63 01 : 71
4E78 20 00 ED B0 3E 01 ED 5B : 44
--------------------------------
SUM: 25 EA DB 19 0E BD 28 B8 6833

4E80 39 63 2A 64 1F CD 03 20 : 39
4E88 D8 CD 43 50 D8 06 10 21 : 47
4E90 26 63 7E FE 7F D2 61 50 : 07
4E98 23 4E E5 2A 62 1F 16 00 : 17
4EA0 5F 19 71 E1 05 CA 3B 4D : 21
4EA8 18 E8 2A 55 60 29 11 D9 : F2
4EB0 4E 19 34 20 1B E5 01 38 : F4
4EB8 00 2A 55 60 CD 46 48 11 : 4B
4EC0 C3 60 19 E5 CD E1 4E E1 : FE
4EC8 38 0D 11 32 00 19 34 E1 : B6
4ED0 7E 23 66 6F 7E B7 C9 E1 : 55
4ED8 C9 00 B8 00 B9 00 BA 00 : F4
4EE0 BB 01 38 00 22 A8 50 11 : 1F
4EE8 06 63 ED B0 3A 37 63 32 : 0C
4EF0 5D 1F 3A 38 63 47 3A 3D : 0F
4EF8 63 B8 DA 3B 4D 78 CD 27 : E9
--------------------------------
SUM: E2 F0 75 3B 35 31 DE 4A 9B51

4F00 50 EB 3A 55 60 C6 B8 67 : 0F
4F08 2E 00 3E 01 CD 00 20 D8 : 32
4F10 01 38 00 ED 5B A8 50 21 : 9A
4F18 06 63 ED B0 B7 C9 2A ED : 9D
4F20 61 23 22 ED 61 2A 48 4F : B5
4F28 77 21 48 4F 34 37 3F C0 : 99
4F30 21 00 BC 22 25 50 21 DB : 70
4F38 61 CD 76 4F D8 21 0D 62 : 5B
4F40 34 C0 36 FF 37 3E 05 C9 : 6C
4F48 00 BC 2A B5 61 23 22 B5 : F6
4F50 61 2A 74 4F 77 21 74 4F : A9
4F58 34 37 3F C0 21 00 BD 22 : 6A
4F60 25 50 21 A3 61 CD 76 4F : 2C
4F68 D8 21 D5 61 34 C0 36 FF : 58
4F70 37 3E 05 C9 00 BD 01 38 : 39
4F78 00 11 06 63 22 A8 50 ED : 81
--------------------------------
SUM: DC 34 15 93 B8 7D 5C FB 250C

4F80 B0 3A 37 63 32 5D 1F 3A : 6C
4F88 38 63 47 3A 3D 63 B8 30 : A4
4F90 76 CD 43 50 D8 3A 38 63 : 83
4F98 E6 F0 47 3A 3D 63 E6 F0 : CD
4FA0 B8 30 47 CB 3F CB 3F CB : 0E
4FA8 3F CB 3F 21 27 63 16 00 : 0A
4FB0 5F 19 E5 3A 3D 63 E6 F0 : 0D
4FB8 CD 27 50 CD 94 50 2A 62 : 81
4FC0 1F 16 00 5F 19 36 8F CD : 3F
4FC8 72 50 77 E1 D8 77 23 36 : C2
4FD0 80 2A 62 1F 16 00 5F 19 : B9
4FD8 36 80 3A 3D 63 E6 F0 C6 : 2C
4FE0 10 32 3D 63 CD 61 50 D8 : 38
4FE8 18 9D 3A 38 63 32 3D 63 : 5C
4FF0 CB 3F CB 3F CB 3F CB 3F : 28
4FF8 21 27 63 16 00 5F 19 3A : 73
--------------------------------
SUM: C2 DA 7B A6 20 02 CC 70 471E

5000 38 63 E6 0F C6 80 77 3A : 87
5008 38 63 CD 27 50 EB 3E 01 : 09
5010 2A 25 50 CD 03 20 D8 01 : 68
5018 38 00 ED 5B A8 50 21 06 : 9F
5020 63 ED B0 B7 C9 00 00 F5 : 75
5028 F5 CB 3F CB 3F CB 3F CB : DE
5030 3F 21 26 63 16 00 5F 19 : 77
5038 7E CD 8C 50 F1 E6 0F 85 : 92
5040 6F F1 C9 D5 E5 3A AA 50 : 17
5048 57 3A 5D 1F BA 28 0F 32 : 30
5050 AA 50 3E 01 ED 5B 5E 1F : FE
5058 2A 62 1F CD 00 20 E1 D1 : 4A
5060 C9 D5 E5 3E 01 ED 5B 5E : 68
5068 1F 2A 62 1F CD 03 20 E1 : 9B
5070 D1 C9 C5 E5 06 80 2A 62 : 56
5078 1F 7E B7 28 08 23 10 F9 : B0
--------------------------------
SUM: 59 B4 D7 BF 38 FC 08 AC 47BD

5080 3E 09 37 18 04 3E 80 90 : E8
5088 A7 E1 C1 C9 26 00 6F 29 : D0
5090 29 29 29 C9 E5 CB 3C CB : FB
5098 1D CB 3C CB 1D CB 3C CB : DE
50A0 1D CB 3C CB 1D 7D E1 C9 : 33
50A8 00 00 00                : 00
--------------------------------
SUM: 48 A9 99 40 49 51 48 18 6934
```

リスト2 WZDソースリスト1

```
 >0000'                      ;=======================================
0000'                        ;
0000'                        ; Relocatable Assembler      on S-OS'SWORD'
0000'                        ;           Programmed by T.Ishigami
0000'                        ;                '90 Mar 8th
0000'                        ;
0000'                        ;=======================================
0000'
0000'                                CSEG
0000'                        ;
0000'                        ; Externals
0000'                        ;
                                     EXT     pass
                                     EXT     endflg
                                     EXT     MUL
                                     EXT     PRDEC
                                     EXT     ERRCNT
0000'
0000'
0000'                     C  INCLUDE  SOS.DEF
1FFA                      C _HOT      EQU    1FFAH
1FF4                      C _PRINT    EQU    1FF4H
1FF1                      C _PRNTS    EQU    1FF1H
1FEE                      C _LTNL     EQU    1FEEH
1FEB                      C _NL       EQU    1FEBH
1FE5                      C _MSX      EQU    1FE5H
1FDF                      C _TAB      EQU    1FDFH
1FD3                      C _GETL     EQU    1FD3H
1FC7                      C _PAUSE    EQU    1FC7H
1FC1                      C _PRTHX    EQU    1FC1H
1FBE                      C _PRTHL    EQU    1FBEH
1FB8                      C _HEX      EQU    1FB8H
1FA3                      C _FILE     EQU    1FA3H
1FAF                      C _WOPEN    EQU    1FAFH
1F9A                      C _POKE     EQU    1F9AH
1F94                      C _PEEK     EQU    1F94H
2009                      C _ROPEN    EQU    2009H
2015                      C _KILL     EQU    2015H
2012                      C _NAME     EQU    2012H
2033                      C _ERROR    EQU    2033H
0000'                     C
2000                      C _DRDSB    EQU    2000H
2003                      C _DWTSB    EQU    2003H
0000'                     C
1F7A                      C _PRCNT    EQU    1F7AH
1F76                      C _KBFAD    EQU    1F76H
1F74                      C _IBFAD    EQU    1F74H
1F72                      C _SIZE     EQU    1F72H
1F6A                      C _MEMAX    EQU    1F6AH
1F64                      C _DTBUF    EQU    1F64H
1F62                      C _FATBF    EQU    1F62H
1F60                      C _DIRPS    EQU    1F60H
1F5E                      C _FATPOS   EQU    1F5EH
1F5D                      C _DSK      EQU    1F5DH
0000'                     C
0000'                     C  INCLUDE  WZD.DEF
0000'                     C ; Header File For WZD
0000'                     C ;    CSEG 3000H-
0000'                     C ;    DSEG 6000H-
0000'                     C ;
0000'                     C
BB00                      C RDBUF     EQU 0BB00H
BC00                      C WRBUF1    EQU 0BC00H
BD00                      C WRBUF2    EQU 0BD00H
0000'                     C
0000'                          ;
0000'                        ; Constants
0000'                        ;
000D                         NAME     EQU    13
0003                         EXT      EQU    3
0050                         MAXLINE  EQU    80
0004                         INCMAX   EQU    4
000D                         CR       EQU    0DH
001B                         BRK      EQU    1BH
0000'
0000'                        ;
0000'                        ; Start
0000'                        ;
0000' ED 73 ** **            main:    LD     (SPBUF),SP
0004' ED 7B 6A 1F                     LD     SP,(_MEMAX)
0008'
0008' ED 5B 76 1F                     LD     DE,(_KBFAD)
000C'                        ;        INC    DE
000C'                        ;        INC    DE          ;Skip '# (WLK)' or '#J(3000)'
000C'
000C' 06 01                           LD     B,1         ;B = argc
000E' 1A                     CC1:     LD     A,(DE)
000F' 13                              INC    DE
0010' A7                              AND    A
0011' 28 07                           JR     Z,CC3
0013' FE 20                           CP     ' '
0015' 20 F7                           JR     NZ,CC1
0017' 04                              INC    B           ;if (DE) = ' ' then argc++
0018' 18 F4                           JR     CC1
001A'
001A' 78                     CC3:     LD     A,B
001B' 32 ** **               LD       (argc),A
001E' AF                     CC5:     XOR    A
001F' 32 ** **               LD       (soname),A
```

▶やはりポピュラスとダンジョンマスターは，編集部のほうで「親の遺言指定（古い）」していただかないと，恐くて夜もおちおち寝られやしません。

中村　隆生（20）神奈川県

```
0022' 32 ** **          LD      (prname),A
0025' 32 ** **          LD      (rename),A
0028' 32 ** **          LD      (wkpr),A
002B' 32 ** **          LD      (wkre),A
002E'
002E' 3A ** **          LD      A,(argc)
0031' FE 01                     CP      1
0033' 20 05                     JR      NZ,CC6
0035' CD ** **          CALL    GETL
0038' 18 0C                     JR      CC7
003A'
003A' 2A 76 1F  CC6:    LD      HL,(_KBFAD)
003D'                   ;       INC     HL
003D'                   ;       INC     HL              ;Skip '# ' or '#J'
003D' 7E                CC4:    LD      A,(HL)
003E' 23                        INC     HL
003F' FE 20                     CP      ' '
0041' 20 FA                     JR      NZ,CC4
0043' 22 ** **          LD      (ptr),HL
0046'
0046' 21 ** **  CC7:    LD      HL,CC2+0
0049' CD ** **          CALL    amatch
004C' 11 ** **          LD      DE,soname
004F' 2A ** **          LD      HL,(ptr)
0052' CC ** **          CALL    Z,flcopy
0055'
0055' 11 ** **          LD      DE,rename
0058' 2A ** **          LD      HL,(ptr)
005B' CD ** **          CALL    flcopy
005E' 22 ** **          LD      (ptr),HL
0061'
0061' 21 ** **          LD      HL,CC2+2
0064' CD ** **          CALL    amatch
0067' 11 ** **          LD      DE,prname
006A' 2A ** **          LD      HL,(ptr)
006D' CC ** **          CALL    Z,flcopy
0070' 22 ** **          LD      (ptr),HL
0073'
0073' 21 ** **          LD      HL,CC2+0
0076' CD ** **          CALL    amatch
0079' 11 ** **          LD      DE,soname
007C' 2A ** **          LD      HL,(ptr)
007F' CC ** **          CALL    Z,flcopy
0082' 22 ** **          LD      (ptr),HL
0085'
0085' 2A ** **  CC11:   LD      HL,(ptr)
0088' 7E                        LD      A,(HL)
0089' A7                        AND     A
008A' CA ** **          JP      Z,CC12
008D'
008D' 21 ** **          LD      HL,strRr
0090' CD ** **          CALL    amatch
0093' 28 08                     JR      Z,CC14
0095' 21 ** **          LD      HL,strRr+3
0098' CD ** **          CALL    amatch
009B' 20 0C                     JR      NZ,CC15
009D'
009D' 11 ** **  CC14:   LD      DE,rename
00A0' 21 ** **          LD      HL,soname
00A3' CD ** **          CALL    flcopy
00A6' C3 ** **          JP      CC11
00A9'
00A9' 21 ** **  CC15:   LD      HL,strLl
00AC' CD ** **          CALL    amatch
00AF' 28 08                     JR      Z,CC16
00B1' 21 ** **          LD      HL,strLl+3
00B4' CD ** **          CALL    amatch
00B7' 20 0C                     JR      NZ,CC19
00B9'
00B9' 11 ** **  CC16:   LD      DE,prname
00BC' 21 ** **          LD      HL,soname
00BF' CD ** **          CALL    flcopy
00C2' C3 ** **          JP      CC11
00C5'
00C5' 21 ** **  CC19:   LD      HL,strUSG
00C8' CD ** **          CALL    puts
00CB' CD ** **          CALL    exit
00CE' C3 ** **          JP      CC11
00D1'
00D1' 3A ** **  CC12:   LD      A,(soname)
00D4' A7                        AND     A
00D5' 11 ** **          LD      DE,soname
00D8' 21 ** **          LD      HL,strASM
00DB' C4 ** **          CALL    NZ,flneat
00DE'
00DE' 3A ** **          LD      A,(rename)
00E1' A7                        AND     A
00E2' 11 ** **          LD      DE,rename
00E5' 21 ** **          LD      HL,strREL
00E8' C4 ** **          CALL    NZ,flneat
00EB'
00EB' 3A ** **          LD      A,(prname)
00EE' A7                        AND     A
00EF' 11 ** **          LD      DE,prname
00F2' 21 ** **          LD      HL,strPRN
00F5' C4 ** **          CALL    NZ,flneat
00F8'
00F8' 11 ** **          LD      DE,soname
00FB' 21 ** **          LD      HL,prname
00FE' CD ** **          CALL    flcmp
0101' 21 ** **          LD      HL,strERR
0104' D4 ** **          CALL    NC,error
0107'
0107' 11 ** **          LD      DE,soname
010A' 21 ** **          LD      HL,rename
010D' CD ** **          CALL    flcmp
0110' 21 ** **          LD      HL,strERR
0113' D4 ** **          CALL    NC,error
0116'
0116' 11 ** **          LD      DE,prname
0119' 21 ** **          LD      HL,rename
011C' CD ** **          CALL    flcmp
011F' 21 ** **          LD      HL,strERR
0122' D4 ** **          CALL    NC,error
0125'
0125' 3A ** **          LD      A,(soname)
0128' A7                        AND     A
0129' 21 ** **          LD      HL,strERR
012C' CC ** **          CALL    Z,error
012F'
012F' CD ** **          CALL    pass1
0132' CD ** **          CALL    pass2
0135'
0135' 2A ** **          LD      HL,(ERRCNT)
0138' 7C                        LD      A,H
0139' B5                        OR      L
013A' F5                        PUSH    AF
013B' C4 ** **          CALL    NZ,PRDEC
013E' F1                        POP     AF
013F' 21 ** **          LD      HL,strNO
0142' CC ** **          CALL    Z,puts
0145' 21 ** **          LD      HL,strFerr
0148' CD ** **          CALL    puts
014B'
014B' 3A ** **          LD      A,(argc)
014E' FE 01                     CP      1
0150' CA ** **          JP      Z,CC5           ;if argc = 1
0153' ED 7B ** **               LD      SP,(SPBUF)
0157' C9                        RET
0158'
0158' 4E 6F 00  strNO:  DB      'No',0
015B' 20 46 61 74 61    strFerr:DB      ' Fatal Error(s)',CR,0
0160' 6C 20 45 72 72
0165' 6F 72 28 73 29
016A' 0D 00
016C'
016C'
016C'                   ;
016C'                   ; 1 Line input from console
016C'                   ;
016C' 3E 2A             GETL:   LD      A,'*'
016E' CD F4 1F          CALL    _PRINT
0171' ED 5B 76 1F               LD      DE,(_KBFAD)
0175' CD D3 1F          CALL    _GETL
0178' 1A                        LD      A,(DE)
0179' FE 13                     CP      BRK
017B' CA ** **          JP      Z,exit
017E'
017E' 13                        INC     DE              ;Skip prompt '*'
017F' 1A                        LD      A,(DE)
0180' A7                        AND     A
0181' 28 E9                     JR      Z,GETL
0183' ED 53 ** **               LD      (ptr),DE
0187' C9                        RET
0188'
0188' 3D 00 2C 00       CC2:    DB      '=',0,',',0
018C' 2F 72 00 2F 52    strRr:  DB      '/r',0,'/R',0
0191' 00
0192' 2F 6C 00 2F 4C    strLl:  DB      '/l',0,'/L',0
0197' 00
0198' 41 53 4D 00       strASM: DB      'ASM',0
019C' 52 45 4C 00       strREL: DB      'REL',0
01A0' 50 52 4E 00       strPRN: DB      'PRN',0
01A4'
01A4' 55 73 61 67 65    strUSG: DB      'Usage: WZD [REL],[PRN]=ASM',CR,0
01A9' 3A 20 57 5A 44
01AE' 20 5B 52 45 4C
01B3' 5D 2C 5B 50 52
01B8' 4E 5D 3D 41 53
01BD' 4D 0D 00
01C0' 53 6F 75 72 63    strERR: DB      'Source file name is not given',CR,0
01C5' 65 20 66 69 6C
01CA' 65 20 6E 61 6D
01CF' 65 20 69 73 20
01D4' 6E 6F 74 20 67
01D9' 69 76 65 6E 0D
01DE' 00
01DF' 41 53 4D 09 66    msgASM: DB      'ASM    file :',0
01E4' 69 6C 65 20 3A
01E9' 00
01EA' 0D 52 45 4C 20    msgREL: DB      CR,'REL file :',0
01EF' 66 69 6C 65 20
01F4' 3A 00
01F6' 0D 50 52 4E 20    msgPRN: DB      CR,'PRN file :',0
01FB' 66 69 6C 65 20
0200' 3A 00
0202'
0202'
0202'                   ;
0202'                   ; PASS-1
0202'                   ;
0202' AF                pass1:  XOR     A
0203' 32 ** **          LD      (pass),A
0206' 32 ** **          LD      (endflg),A
0209' CD ** **          CALL    inihsh
020C' CD ** **          CALL    iniwk1
020F' 21 00 00          LD      HL,0
0212' 22 ** **          LD      (flvl),HL       ;Clear Include Level
0215'
0215' 3E 04                     LD      A,4             ;ASC
0217' 11 ** **          LD      DE,soname
021A' 21 ** **          LD      HL,wkso
021D' CD ** **          CALL    RDOPEN
0220' 21 ** **          LD      HL,soname
0223' DA ** **          JP      C,openerr
0226' 3E FF                     LD      A,0FFH
0228' 32 ** **          LD      (RDPNT),A       ;Clear POINTER for reading
022B'
022B' CD ** **  CC35:   CALL    getlin
022E' 38 0B                     JR      C,CC36
0230' 3A ** **          LD      A,(endflg)
0233' A7                        AND     A
0234' 20 05                     JR      NZ,CC36
0236'
0236' CD ** **          CALL    linasm
0239' 18 F0                     JR      CC35
023B'
023B' 3A ** **  CC36:   LD      A,(flvl)
023E' 3D                        DEC     A
023F' 32 ** **          LD      (flvl),A
0242' 3C                        INC     A
0243' C2 ** **          JP      NZ,CC35
0246' C9                        RET
0247'
0247'
0247' 3E 01             pass2:  LD      A,1
0249' 32 ** **          LD      (pass),A
024C' AF                        XOR     A
024D' 32 ** **          LD      (endflg),A
0250' 32 ** **          LD      (flvl),A
0253' CD ** **          CALL    iniwk2
0256'
0256' 3E 04                     LD      A,4             ;ASCII FILE
0258' 11 ** **          LD      DE,soname
025B' 21 ** **          LD      HL,wkso
025E' CD ** **          CALL    RDOPEN
0261' 21 ** **          LD      HL,soname
0264' DC ** **          CALL    C,openerr
0267' 3E FF                     LD      A,0FFH
0269' 32 ** **          LD      (RDPNT),A       ;Clear Pointer for reading
026C'
026C' 3A ** **          LD      A,(rename)
026F' A7                        AND     A
0270' 28 15                     JR      Z,CC43
0272'
0272' 3E 01                     LD      A,1             ;BINARY FILE
0274' 11 ** **          LD      DE,rename
0277' 21 ** **          LD      HL,wkre
027A' CD ** **          CALL    WROPEN
027D' 21 ** **          LD      HL,rename
0280' DA ** **          JP      C,openerr
0283' AF                        XOR     A
0284' 32 ** **          LD      (WRPNT1),A      ;Clear POINTER1 for writing
0287'
0287' 3A ** **  CC43:   LD      A,(prname)
028A' 32 ** **          LD      (LSTSW),A       ;LSTSW off (if prname is not present
ed)
028D' A7                        AND     A
028E' 28 15                     JR      Z,CC45
0290'
0290' 3E 04                     LD      A,4             ;ASCII file
0292' 11 ** **          LD      DE,prname
0295' 21 ** **          LD      HL,wkpr
0298' CD ** **          CALL    WROPEN
029B' 21 ** **          LD      HL,prname
029E' DA ** **          JP      C,openerr
02A1' AF                        XOR     A
02A2' 32 ** **          LD      (WRPNT2),A      ;Clear POINTER2 for writing
02A5'
02A5' CD ** **  CC45:   CALL    getlin
02A8' 38 0B                     JR      C,CC46
02AA' 3A ** **          LD      A,(endflg)
02AD' A7                        AND     A
02AE' 20 05                     JR      NZ,CC46
02B0'
02B0' CD ** **          CALL    linasm
02B3' 18 F0                     JR      CC45
02B5'
02B5' 3A ** **  CC46:   LD      A,(flvl)
02B8' 3D                        DEC     A
02B9' 32 ** **          LD      (flvl),A
02BC' 3C                        INC     A
02BD' C2 ** **          JP      NZ,CC45                 ;if (flvl)-- != 0  JP CC45
02C0'
02C0' 3A ** **          LD      A,(rename)
02C3' A7                        AND     A
02C4' 28 14                     JR      Z,CC47
02C6'
02C6' CD ** **          CALL    PUTbF
02C9' 3E FF                     LD      A,0FFH
02CB' CD ** **          CALL    putrel
02CE' 21 00 BC          LD      HL,WRBUF1
02D1' 22 ** **          LD      (BUFAD),HL
02D4' 21 ** **          LD      HL,wkre
02D7' CD ** **          CALL    CLOSE
02DA'
02DA' 3A ** **  CC47:   LD      A,(prname)
02DD' A7                        AND     A
02DE' C8                        RET     Z
02DF' 3E 00                     LD      A,0
02E1' CD ** **          CALL    putprn          ;PUT END CODE into PRN-file
02E4' 21 00 BD          LD      HL,WRBUF2
02E7' 22 ** **          LD      (BUFAD),HL
02EA' 21 ** **          LD      HL,wkpr
02ED' CD ** **          CALL    CLOSE
02F0' C9                        RET
02F1'
02F1'                   ;
02F1'                   ; Include Routine called from 'linasm'
02F1'                   ;
02F1'                   include::
02F1' ED 53 ** **               LD      (fptr),DE
02F5'
02F5' 3A ** **          LD      A,(flvl)
02F8' 3C                        INC     A
02F9' 32 ** **          LD      (flvl),A
02FC' FE 04                     CP      INCMAX
02FE' 38 09                     JR      C,CC50
0300' 21 ** **          LD      HL,CC49+0
0303' CD ** **          CALL    puts
0306' C3 ** **          JP      exit
0309'
0309' 2A ** **  CC50:   LD      HL,(flvl)
030C' 01 12 00          LD      BC,NAME + 1 + EXT + 1
030F' CD ** **          CALL    MUL
0312' 11 ** **          LD      DE,soname
0315' 19                        ADD     HL,DE
0316' 22 ** **          LD      (file),HL
0319'
0319' ED 5B ** **               LD      DE,(fptr)
031D' 1A                CC53:   LD      A,(DE)
031E' CD ** **          CALL    isflchr
0321' 38 05                     JR      C,CC52
0323' 13                        INC     DE
0324' 77                        LD      (HL),A
0325' 23                        INC     HL
0326' 18 F5                     JR      CC53
0328'
0328' 36 00             CC52:   LD      (HL),0
032A' ED 53 ** **               LD      (fptr),DE
032E'
032E' 01 38 00          LD      BC,IMF_SIZE
0331' 2A ** **          LD      HL,(flvl)
0334' CD ** **          CALL    MUL
0337' 11 ** **          LD      DE,wkso
033A' 19                        ADD     HL,DE           ;HL = IMF_SIZE * (flvl) + wks
o
033B'
033B' 3E 04                     LD      A,4             ;ASCII file
033D' ED 5B ** **               LD      DE,(file)
0341' CD ** **          CALL    RDOPEN
0344' 2A ** **          LD      HL,(file)
0347' DA ** **          JP      C,openerr
034A'
034A' 2A ** **          LD      HL,(flvl)
034D' 29                        ADD     HL,HL
034E' 11 ** **          LD      DE,RDPNT
0351' 19                        ADD     HL,DE
0352' 36 FF                     LD      (HL),0FFH       ;Clear respective POINTER for
reading
0354'
0354' ED 5B ** **               LD      DE,(fptr)
0358' C9                        RET
0359'
0359' 49 4E 43 4C 55    CC49:   DB      'INCLUDE level over ',CR
035E' 44 45 20 6C 65
0363' 76 65 6C 20 6F
0368' 76 65 72 20 0D
036D'
036D'
036D'                   fptr:   DS      2
036F'                   file:   DS      2
0371'
0371' E5                openerr:PUSH    HL
0372' 21 ** **          LD      HL,CC56
0375' CD ** **          CALL    puts
0378' E1                        POP     HL
0379' CD ** **          CALL    puts
037C' CD EE 1F          CALL    _LTNL
037F' C3 ** **          JP      exit
0382'
0382' 43 61 6E 20 6E    CC56:   DB      "Can not open ",0
0387' 6F 74 20 6F 70
038C' 65 6E 20 00
0390'
0390' 06 50             getlin: LD      B,MAXLINE
0392' 21 ** **          LD      HL,strbuf
0395'
0395' C5                CC58:   PUSH    BC
0396' E5                        PUSH    HL
0397' CD ** **          CALL    INPUT_
039A' E1                        POP     HL
039B' C1                        POP     BC
039C'
039C' A7                        AND     A
039D' 28 0B                     JR      Z,CC63
039F' FE 0D                     CP      CR
03A1' 28 0B                     JR      Z,CC66
03A3'
03A3' 77                        LD      (HL),A
03A4' 23                        INC     HL
03A5' 05                        DEC     B
03A6' 20 ED                     JR      NZ,CC58
03A8' B7                        OR      A               ;CY = 0
03A9' C9                        RET
03AA'
03AA' AF                CC63:   XOR     A
03AB' 77                        LD      (HL),A
03AC' 37                        SCF
03AD' C9                        RET
03AE'
03AE' 3E 0D             CC66:   LD      A,CR
03B0' 77                        LD      (HL),A
03B1' 23                        INC     HL
03B2' AF                        XOR     A
03B3' 77                        LD      (HL),A
03B4' C9                        RET                     ;CY = 0
03B5'
03B5' E5                putrel::PUSH    HL
03B6' 21 ** **          LD      HL,rename
03B9' 6E                        LD      L,(HL)
03BA' 2C                        INC     L
03BB' 2D                        DEC     L
03BC' E1                        POP     HL
03BD' C8                        RET     Z               ;if (rename) = 0 RET
03BE'
03BE' CD ** **          CALL    PRINTr
03C1' D0                        RET     NC
03C2' CD 33 20          CALL    _ERROR
03C5' C3 ** **          JP      exit
03C8'
03C8'
03C8' E5                putprn::PUSH    HL
03C9' B7                        OR      A               ;CY = 0
03CA' 21 ** **          LD      HL,LSTSW
03CD' 6E                        LD      L,(HL)
03CE' 2C                        INC     L
03CF' 2D                        DEC     L
03D0' 28 07                     JR      Z,putprl
03D2'
03D2' D5                        PUSH    DE
03D3' C5                        PUSH    BC
03D4'
03D4' CD ** **          CALL    PRINTp
03D7'
03D7' C1                        POP     BC
03D8' D1                        POP     DE
03D9' E1                putprl: POP     HL
03DA' D0                        RET     NC
03DB' CD 33 20          CALL    _ERROR
03DE' C3 ** **          JP      exit
03E1'
03E1' ED 5B ** **       amatch: LD      DE,(ptr)
03E5'
03E5' 7E                CC70:   LD      A,(HL)
03E6' A7                        AND     A
03E7' 28 07                     JR      Z,CC71
03E9' 1A                        LD      A,(DE)
03EA' BE                        CP      (HL)
03EB' C0                        RET     NZ              ;RET with NZ
03EC' 13                        INC     DE
03ED' 23                        INC     HL
03EE' 18 F5                     JR      CC70
03F0'
03F0' ED 53 ** **       CC71:   LD      (ptr),DE
03F4' C9                        RET                     ;Z = 1
03F5'
03F5' 06 0D             flneat: LD      B,NAME          ;B = counter
03F7'                   CC75:
03F7' 1A                        LD      A,(DE)
03F8' CD ** **          CALL    isflchr
03FB' 38 0A                     JR      C,CC76
03FD' 05                        DEC     B
03FE' 28 07                     JR      Z,CC76
0400' FE 2E                     CP      '.'
0402' 28 03                     JR      Z,CC76
0404' 13                        INC     DE
0405' 18 F0                     JR      CC75
0407'
0407' 1A                CC76:   LD      A,(DE)
0408' FE 2E                     CP      '.'
040A' C8                        RET     Z
040B' 3E 2E                     LD      A,'.'
040D' 12                        LD      (DE),A
040E' 13                        INC     DE
040F' CD ** **          CALL    flcopy
0412' C9                        RET
0413'
0413' 06 0D             flcopy: LD      B,NAME          ;B = counter
0415' AF                        XOR     A
0416' 12                        LD      (DE),A
0417'                   CC82:
0417' 7E                        LD      A,(HL)
0418' CD ** **          CALL    isflchr
041B' D8                        RET     C
041C' 05                        DEC     B
041D' C8                        RET     Z
041E'
041E' 7E                        LD      A,(HL)
041F' 23                        INC     HL
0420' 12                        LD      (DE),A
0421' 13                        INC     DE
0422' AF                        XOR     A
0423' 12                        LD      (DE),A
0424' 18 F1                     JR      CC82
0426'
```

THE SENTINEL

▶暴れん坊将軍IIIの「この子　どこの子　め組の子」というサブタイトルは，日立の「この木　なんの木」と関係あるのでしょうか。　　坊農　誠（19）福井県

```
0426' 1A                flcmp:  LD      A,(DE)
0427' A7                        AND     A
0428' 37                        SCF
0429' C8                        RET     Z               ;CY = 1
042A' 7E                        LD      A,(HL)
042B' A7                        AND     A
042C' 37                        SCF
042D' C8                        RET     Z               ;CY = 1
042E'                   CC90:
042E' 1A                        LD      A,(DE)
042F' A7                        AND     A
0430' C8                        RET     Z               ;CY = 0
0431' 13                        INC     DE
0432' BE                        CP      (HL)
0433' 23                        INC     HL
0434' 37                        SCF
0435' C0                        RET     NZ              ;CY = 1
0436' 18 F6                     JR      CC90
0438'
0438' CD ** **  error:  CALL    puts
043B' CD ** **          CALL    exit
043E' C9                        RET
043F'
043F' FE 3D             isflchr:CP      '='
0441' 28 0A                     JR      Z,isfl1
0443' FE 2F                     CP      '/'
0445' 28 06                     JR      Z,isfl1
0447' FE 2C                     CP      ','
0449' 28 02                     JR      Z,isfl1
044B' A7                        AND     A
044C' C0                        RET     NZ              ;CY = 0
044D' 37                isfl1:  SCF
044E' C9                        RET                     ;CY = 1
044F'
044F' 7E                puts:   LD      A,(HL)
```

```
0450' 23                        INC     HL
0451' A7                        AND     A
0452' C8                        RET     Z
0453' E5                        PUSH    HL
0454' CD F4 1F          CALL    _PRINT
0457' E1                        POP     HL
0458' 18 F5                     JR      puts
045A'
045A' 3A ** **  exit:   LD      A,(wkre)
045D' A7                        AND     A
045E' 21 00 BC          LD      HL,WRBUF1
0461' 22 ** **          LD      (BUFAD),HL
0464' 21 ** **          LD      HL,wkre
0467' C4 ** **          CALL    NZ,CLOSE
046A'
046A' 3A ** **          LD      A,(wkpr)
046D' A7                        AND     A
046E' 21 00 BD          LD      HL,WRBUF2
0471' 22 ** **          LD      (BUFAD),HL
0474' 21 ** **          LD      HL,wkpr
0477' C4 ** **          CALL    NZ,CLOSE
047A'
047A' ED 7B ** **               LD      SP,(SPBUF)
047E' C9                        RET
047F'
047F'                   ;
047F'                   ; Works
047F'                   ;
047F'                           DSEG
0000"
0000"                   WKstart::               ;The Mark Of the Starting Wor
k Area
0000"
0000"                   SPBUF:  DS      2       ;Buffer of The SP reg.
```

```
0002"                   argc:   DS      1               ;Counter of The Arguement
0003"                   ptr:    DS      2
0005"                   strbuf::DS      MAXLINE
0055"                   flvl::  DS      2               ;Include File leval
0057"                   soname: DS      (NAME+1+EXT+1) * INCMAX
005B"                   prname::DS      (NAME+1+EXT+1)
006D"                   rename::DS      (NAME+1+EXT+1)
007F"
0038                    IMF_SIZE EQU    38H
007F"                   wkso::  DS      IMF_SIZE * INCMAX
0083"                   wkpr::  DS      IMF_SIZE
00BB"                   wkre::  DS      IMF_SIZE
00F3"
00F3"                   ;
00F3"                   ; Externals
00F3"                   ;
                                EXT     RDOPEN
                                EXT     WROPEN
                                EXT     WRPNT1
                                EXT     WRPNT2
                                EXT     RDPNT
                                EXT     CLOSE
                                EXT     PRINTr
                                EXT     PRINTp
                                EXT     INPUT_
                                EXT     BUFAD
00F3"
                                EXT     PUTbF
                                EXT     iniwk1
                                EXT     iniwk2
                                EXT     inihsh
                                EXT     linasm
                                EXT     LSTSW
00F3"                           END
```

リスト3 WZDソースリスト2

```
0000'                   ;================================
0000'                   ;
0000'                   ; WZD part II
0000'                   ;  Relocatable Assembler For S-OS
0000'                   ;       Programed by T.Ishigami
0000'                   ;               '87 Nov 26th
0000'                   ;
0000'                   ;================================
0000'
0000'                           CSEG
0000'                   ;
0000'                   ; Externals
0000'                   ;
                                EXT     CNTbA
                                EXT     CNTeA
                                EXT     BFOBJ
                                EXT     BFEVAL
                                EXT     INCPC
                                EXT     pass
                                EXT     UNDEF
                                EXT     LSTOBJ
                                EXT     SEGMD
                                EXT     prname
                                EXT     strbuf
                                EXT     GETADRS
                                EXT     flvl
                                EXT     PAGE
                                EXT     CNTLINE
                                EXT     ERRNUM
                                EXT     TITLBF
                                EXT     putprn
                                EXT     putrel
0000'
0000'
0009                    TAB     EQU     09H
000D                    CR      EQU     0DH
0000'                   ;
0000'                   ;== Subroutines to  Put Object Code ==
0000'                   ;
0000' CD ** **  PUToA:: CALL    INCPC           ;Increment respectable PC
0003'
0003' E5                        PUSH    HL
0004' 21 ** **          LD      HL,pass
0007' 6E                        LD      L,(HL)
0008' 2D                        DEC     L
0009' E1                        POP     HL
000A' C0                        RET     NZ              ;When pass-1 ,ret
000B'
000B' F5                        PUSH    AF
000C' 3A ** **          LD      A,(prname)
000F' A7                        AND     A
0010' 28 08                     JR      Z,PUTo1
0012'
0012' F1                        POP     AF
0013' F5                        PUSH    AF
0014' CD ** **          CALL    pPRTHX
0017' CD ** **          CALL    pPRNTS
001A'
001A' 3A ** **  PUTo1:  LD      A,(LSTOBJ)
001D' FE 60                     CP      60H
001F' 20 20                     JR      NZ,PUTo2
0021' F1                        POP     AF
0022'
0022' D5                        PUSH    DE
0023' E5                        PUSH    HL
0024' ED 5B ** **               LD      DE,(CNTbA)
0028' 16 00                     LD      D,0             ;LD E,(CNTbA)
002A' 21 ** **          LD      HL,BFOBJ
002D' 19                        ADD     HL,DE
002E' 77                        LD      (HL),A
002F' 7B                        LD      A,E
0030' E1                        POP     HL
0031' D1                        POP     DE
0032' 3C                        INC     A
0033' E6 1F                     AND     1FH
0035' 32 ** **          LD      (CNTbA),A
0038' C0                        RET     NZ
0039'
0039' C5                        PUSH    BC
003A' 06 20                     LD      B,32            ;if A = 0
003C' CD ** **          CALL    PUTb2
003F' C1                        POP     BC
0040' C9                        RET
0041'
0041'
0041'
0041' C5                PUTo2:  PUSH    BC
0042' 3A ** **          LD      A,(CNTbA)
0045' 47                        LD      B,A
0046' 3A ** **          LD      A,(LSTOBJ)
0049' A7                        AND     A
004A' C4 ** **          CALL    NZ,PUTb2
004D' C1                        POP     BC
004E' 3E 60                     LD      A,60H
0050' 32 ** **          LD      (LSTOBJ),A
0053' 3E 01                     LD      A,1
0055' 32 ** **          LD      (CNTbA),A
0058' F1                        POP     AF
0059' 32 ** **          LD      (BFOBJ+0),A
005C' C9                        RET
005D'
005D'                   ;
005D'                   ; PUT relocatable-ITEM in A reg.
005D'                   ;
005D' C5                PUTrA:: PUSH    BC
005E'
005E' 06 01                     LD      B,1
0060' FE 84                     CP      84H
0062' 28 10                     JR      Z,PUTr1
0064' FE E4                     CP      0E4H
0066' 28 0C                     JR      Z,PUTr1
0068'
0068' 06 02                     LD      B,2
006A' FE 82                     CP      82H
006C' 28 06                     JR      Z,PUTr1
006E' FE E5                     CP      0E5H
0070' 28 02                     JR      Z,PUTr1
0072' 18 0E                     JR      PUTr3
0074'
0074' F5                PUTr1:  PUSH    AF
0075' 3A ** **          LD      A,(prname)
0078' A7                        AND     A
0079' C4 ** **          CALL    NZ,pPRTXX
007C' CD ** **  PUTr2:  CALL    INCPC
007F' 10 FB                     DJNZ    PUTr2
0081' F1                        POP     AF
0082'
```

```
0082' C1                PUTr3:  POP     BC
0083'
0083' E5                PUTr4:  PUSH    HL
0084' 21 ** **          LD      HL,pass
0087' 6E                        LD      L,(HL)
0088' 2D                        DEC     L
0089' E1                        POP     HL
008A' C0                        RET     NZ
008B'
008B' F5                        PUSH    AF              ;For PASS-2
008C' C5                        PUSH    BC
008D' CD ** **          CALL    PUTbF
0090' C1                        POP     BC
0091' F1                        POP     AF
0092' CD ** **          CALL    PUTcA
0095' C9                        RET
0096'                   ;
0096'                   PUTrAHL::
0096' CD ** **          CALL    PUTrA
0099' F5                PUTrHL::PUSH    AF
009A' 7D                        LD      A,L
009B' CD ** **          CALL    PUTr4
009E' 7C                        LD      A,H
009F' CD ** **          CALL    PUTr4
00A2' F1                        POP     AF
00A3' C9                        RET
00A4'
00A4'                   ;
00A4'                   ; PUT INTO BUFFER
00A4'                   ;       in A = Atribute
00A4'                   ;          C = CODE
00A4' CD ** **  PUTbA:: CALL    INCPC           ;does not use ITEM 98h & A0h
00A7'                                           ;so all ITEMs occupy 1 Byte
00A7' E5                        PUSH    HL
00A8' 21 ** **          LD      HL,pass
00AB' 6E                        LD      L,(HL)
00AC' 2D                        DEC     L
00AD' E1                        POP     HL
00AE' C0                        RET     NZ              ;RET when PASS-1
00AF'
00AF' E5                        PUSH    HL
00B0' 21 ** **          LD      HL,LSTOBJ
00B3' BE                        CP      (HL)
00B4' E1                        POP     HL
00B5' 20 1A                     JR      NZ,PUTb1
00B7'
00B7' D5                        PUSH    DE
00B8' E5                        PUSH    HL
00B9' 3A ** **          LD      A,(CNTbA)
00BC' 16 00                     LD      D,0
00BE' 5F                        LD      E,A
00BF' 21 ** **          LD      HL,BFOBJ
00C2' 19                        ADD     HL,DE
00C3' 71                        LD      (HL),C
00C4' E1                        POP     HL
00C5' D1                        POP     DE
00C6' 3C                        INC     A
00C7' E6 07                     AND     07H
00C9' 32 ** **          LD      (CNTbA),A
00CC' C0                        RET     NZ
00CD' 06 08                     LD      B,8             ;if A = 0
00CF' 18 1A                     JR      PUTb2
00D1'
00D1' F5                PUTb1:  PUSH    AF
00D2' 3A ** **          LD      A,(CNTbA)
00D5' 47                        LD      B,A
00D6' 3A ** **          LD      A,(LSTOBJ)
00D9' A7                        AND     A
00DA' C4 ** **          CALL    NZ,PUTb2
00DD' F1                        POP     AF
00DE' 32 ** **          LD      (LSTOBJ),A
00E1' 3E 01                     LD      A,1
00E3' 32 ** **          LD      (CNTbA),A
00E6' 79                        LD      A,C
00E7' 32 ** **          LD      (BFOBJ+0),A
00EA' C9                        RET
00EB'
00EB' 04                PUTb2:  INC     B
00EC' 05                        DEC     B
00ED' C8                        RET     Z               ;if B = 0
00EE'
00EE' 3A ** **          LD      A,(LSTOBJ)
00F1' 80                        ADD     A,B
00F2' 3D                        DEC     A
00F3' CD ** **          CALL    PUTcA
00F6' D5                        PUSH    DE
00F7' E5                        PUSH    HL
00F8' 21 ** **          LD      HL,BFOBJ
00FB' 7E                PUTb3:  LD      A,(HL)
00FC' CD ** **          CALL    PUTcA
00FF' 23                        INC     HL
0100' 10 F9                     DJNZ    PUTb3
0102' E1                        POP     HL
0103' D1                        POP     DE
0104' C9                        RET
0105'                   ;
0105'                   ; BFOBJ ノ CODE ヲ ハカセル
0105'                   ;
0105' 3A ** **  PUTbF:: LD      A,(CNTbA)
0108' 47                        LD      B,A
0109' CD ** **          CALL    PUTb2
010C' AF                        XOR     A
010D' 32 ** **          LD      (CNTbA),A
0110' C9                        RET
0111'                   ;
0111'                   ; PUT Binary Code in Areg.
0111'                   ;
0111'                   PUTcA:
0111' E5                        PUSH    HL
0112' D5                        PUSH    DE
0113' C5                        PUSH    BC
0114' F5                        PUSH    AF
0115' CD ** **          CALL    putrel
0118' F1                        POP     AF
0119' C1                        POP     BC
011A' D1                        POP     DE
011B' E1                        POP     HL
011C' C9                        RET
011D'
011D'
011D'                   ;
011D'                   ; Put Routines For EVAL-buffer
011D'                   ;
011D' E5                PUTeA:: PUSH    HL
011E' 21 ** **          LD      HL,pass
0121' 6E                        LD      L,(HL)
0122' 2D                        DEC     L
```

```
0123' E1                        POP     HL
0124' C0                        RET     NZ              ;RET When PASS-1
0125'
0125' D5                        PUSH    DE
0126' E5                        PUSH    HL
0127' F5                        PUSH    AF
0128' 3A ** **          LD      A,(CNTeA)
012B' 6F                        LD      L,A
012C' 3C                        INC     A
012D' 32 ** **          LD      (CNTeA),A
0130' 26 00                     LD      H,0
0132' 11 ** **          LD      DE,BFEVAL
0135' 19                        ADD     HL,DE
0136' F1                        POP     AF
0137' 77                        LD      (HL),A
0138' E1                        POP     HL
0139' D1                        POP     DE
013A' C9                        RET
013B'
013B'                   PUTeAHL::
013B' CD ** **          CALL    PUTeA
013E' F5                PUTeHL::PUSH    AF
013F' 7D                        LD      A,L
0140' CD ** **          CALL    PUTeA
0143' 7C                        LD      A,H
0144' CD ** **          CALL    PUTeA
0147' F1                        POP     AF
0148' C9                        RET
0149'                   ;
0149'                   ; Flush BFEVAL
0149'                   ;
0149' 3A ** **  PUTeF:: LD      A,(CNTeA)
014C' 21 ** **          LD      HL,BFEVAL
014F'
014F' A7                PUTe1:  AND     A
0150' C8                        RET     Z
0151' F5                        PUSH    AF
0152' 7E                        LD      A,(HL)
0153' 23                        INC     HL
0154' CD ** **          CALL    PUTr4
0157' F1                        POP     AF
0158' 3D                        DEC     A
0159' 18 F4                     JR      PUTe1
015B'
015B' 3A ** **  PUTe82::LD      A,(UNDEF)
015E' E6 01                     AND     1
0160' 20 09                     JR      NZ,Pe821
0162'
0162' 79                        LD      A,C
0163' CD ** **          CALL    PUToA
0166' 78                        LD      A,B
0167' CD ** **          CALL    PUToA
016A' C9                        RET
016B'
016B' CD ** **  Pe821:  CALL    PUTeF
016E' 3E 82                     LD      A,82H
0170' CD ** **          CALL    PUTrA
0173' C9                        RET
0174'
0174'
0174' 3A ** **  PUTe84::LD      A,(UNDEF)
0177' E6 01                     AND     1
0179'
0179' 79                        LD      A,C
017A' CA ** **          JP      Z,PUToA
017D'
017D' CD ** **          CALL    PUTeF
0180' 3E 84                     LD      A,84H
0182' CD ** **          CALL    PUTrA
0185' C9                        RET
0186'
0186'                   ;
0186'                   ; SUB-Routines to make PRN-File
0186'                   ;       Programed By T.Ishigami
0186'                   ;               '89 Nov.1st
0186'                   ;
0186' 3A ** **  PUTHD:: LD      A,(pass)
0189' A7                        AND     A
018A' C8                        RET     Z
018B'                                           ;when PASS-2
018B' 3A ** **          LD      A,(prname)
018E' A7                        AND     A
018F' C8                        RET     Z
0190'                                           ;Only when PRN-name is presen
ted
0190' AF                        XOR     A
0191' 32 ** **          LD      (CURX),A
0194' 3E 01                     LD      A,1
0196' 32 ** **          LD      (TXTFLAG),A
0199' C3 ** **          JP      pPRINT2
019C'
019C'
019C' 3A ** **  PUTHD2::LD      A,(pass)
019F' A7                        AND     A
01A0' C8                        RET     Z
01A1'                                           ;when PASS-2
01A1' 3A ** **          LD      A,(prname)
01A4' A7                        AND     A
01A5' C8                        RET     Z
01A6'                                           ;Only when PRN-name is presen
ted
01A6' AF                        XOR     A
01A7' 32 ** **          LD      (CURX),A
01AA'
01AA' CD ** **          CALL    pPRNTS
01AD' CD ** **          CALL    pPRTHL
01B0' CD ** **          CALL    pPRNTS
01B3' CD ** **          CALL    pPRNTS
01B6' C9                        RET
01B7'
01B7' 3A ** **  PUTHD3::LD      A,(pass)
01BA' A7                        AND     A
01BB' C8                        RET     Z
01BC'                                           ;When PASS-2
01BC' 3A ** **          LD      A,(prname)
01BF' A7                        AND     A
01C0' C8                        RET     Z
01C1'                                           ;Only when PRN-name is presen
ted
01C1' E5                        PUSH    HL
01C2' 21 20 20          LD      HL,2020H
01C5' 22 ** **          LD      (PRNBUF+1),HL
01C8' 22 ** **          LD      (PRNBUF+3),HL
01CB' 22 ** **          LD      (PRNBUF+5),HL
01CE' 3E 07                     LD      A,7
01D0' 32 ** **          LD      (CURX),A
01D3' E1                        POP     HL
```


▶「日刊にしたら，月16,800円ですよ，ホントに買ってくれますか」にひと言。いまと同じ内容で，月30回前後編集できますか。　　小板橋　敏弘（20）福島県

```
01D4' C9                        RET
01D5'
01D5'
01D5'
01D5' F5              pPRINT:   PUSH    AF
01D6' C5                        PUSH    BC
01D7' D5                        PUSH    DE
01D8' E5                        PUSH    HL
01D9'
01D9' 11 ** **        LD        DE,PRNBUF
01DC' 2A ** **        LD        HL,(CURX)
01DF' 26 00                     LD      H,0
01E1' 19                        ADD     HL,DE
01E2' 77                        LD      (HL),A
01E3' 23                        INC     HL
01E4' 36 00                     LD      (HL),0
01E6'
01E6' 3A ** **        LD        A,(CURX)
01E9' 3C                        INC     A
01EA' 32 ** **        LD        (CURX),A
01ED' FE 16                     CP      22
01EF' D4 ** **        CALL      NC,pPRINT1
01F2'
01F2' E1                        POP     HL
01F3' D1                        POP     DE
01F4' C1                        POP     BC
01F5' F1                        POP     AF
01F6' C9                        RET
01F7'
01F7' CD ** ** pPRINT1:CALL     PUTFT
01FA' CD ** ** pPRINT2:CALL     pPRNTS
01FD' CD ** **        CALL      GETADRS
0200' CD ** **        CALL      pPRTHL
0203' CD ** **        CALL      pPRTSG
0206' CD ** **        CALL      pPRNTS
0209' C9                        RET
020A'
020A'
020A'                 ; flush PRN-buffer
020A' 3A ** ** PUTFT::  LD      A,(pass)
020D' A7                        AND     A
020E' C8                        RET     Z
020F'                                                   ;When PASS-2
020F' 3A ** **        LD        A,(prname)
0212' A7                        AND     A
0213' C8                        RET     Z
0214'                                                   ;only when PRN-name is presen
ted
0214' 2A ** **        LD        HL,(CNTLINE)
0217' 7C                        LD      A,H
0218' B5                        OR      L
0219' CC ** **        CALL      Z,NXTPG                 ;Open The new Page
021C'
021C' 2A ** **        LD        HL,(CNTLINE)
021F' 23                        INC     HL
0220' 22 ** **        LD        (CNTLINE),HL
0223' 2B                        DEC     HL
0224' D5                        PUSH    DE
0225' ED 5B ** **               LD      DE,(PAGE)
0229' B7                        OR      A
022A' ED 52                     SBC     HL,DE
022C' D1                        POP     DE
022D' 38 06                     JR      C,pPRT2
022F'
022F' 21 00 00        LD        HL,0
0232' 22 ** **        LD        (CNTLINE),HL ;The last line
0235'
0235' CD ** ** pPRT2:   CALL    PUTEM        ;PUT Error mark (if exists)
0238'
0238' 21 ** **        LD        HL,PRNBUF
023B' 7E              pPRT3:    LD      A,(HL)
023C' A7                        AND     A
023D' 28 06                     JR      Z,pPRT4
023F' 23                        INC     HL
0240' CD ** **        CALL      putprn
0243' 18 F6                     JR      pPRT3
0245'
0245' 3A ** ** pPRT4:   LD      A,(TXTFLAG)
0248' A7                        AND     A
0249' 28 05                     JR      Z,pPRT5
024B' CD ** **        CALL      puttxt
024E' 18 05                     JR      pPRT6
0250'
0250' 3E 0D           pPRT5:    LD      A,CR
0252' CD ** **        CALL      putprn
0255' AF              pPRT6:    XOR     A
0256' 32 ** **        LD        (CURX),A
0259' 32 ** **        LD        (TXTFLAG),A
025C' C9                        RET
025D'
025D'                 ;Open Next Page
025D' 21 00 00 NXTPG:   LD      HL,0
0260' 22 ** **        LD        (CNTLINE),HL
0263'
0263' 3A ** **        LD        A,(TITLBF)
0266' A7                        AND     A
0267' C8                        RET     Z
0268'
0268' 21 ** **        LD        HL,TITLBF
026B' 7E              NXT1:     LD      A,(HL)
026C' 23                        INC     HL
026D' A7                        AND     A
026E' 28 05                     JR      Z,NXT2
0270' CD ** **        CALL      putprn
0273' 18 F6                     JR      NXT1
0275' 3E 0D           NXT2:     LD      A,CR
0277' CD ** **        CALL      putprn
027A'
027A' 21 01 00        LD        HL,1
027D' 22 ** **        LD        (CNTLINE),HL
0280' C9                        RET
0281'
0281'                 ;Put ERR mark
0281' 3A ** ** PUTEM:   LD      A,(ERRNUM)
0284' A7                        AND     A
0285' C8                        RET     Z
0286' 21 ** **        LD        HL,ERRTBL
0289' D5                        PUSH    DE
028A' 16 00                     LD      D,0
028C' 5F                        LD      E,A
028D' 19                        ADD     HL,DE
028E' 7E                        LD      A,(HL)       ;Get Err mark from ERRTBL
028F' D1                        POP     DE
0290' 32 ** **        LD        (PRNBUF),A   ;Put Err mark into PRN-buffer
0293' AF                        XOR     A
0294' 32 ** **        LD        (ERRNUM),A   ;Clear (ERRNUM)
0297' C9                        RET
0298'
0298' 53 55 4D 53 53  ERRTBL:   DB      'SUMSSASEA'
029D' 41 53 45 41
02A1' 53 53 53 49 41            DB      'SSSIAVRI'
02A6' 56 52 49
02A9'
02A9' F5              pPRTHX: PUSH      AF
02AA' 0F                        RRCA
02AB' 0F                        RRCA
02AC' 0F                        RRCA
02AD' 0F                        RRCA
02AE' CD ** **        CALL      prthx1
02B1' F1                        POP     AF
02B2' E6 0F           prthx1:   AND     0FH
02B4' C6 30                     ADD     A,'0'
02B6' FE 3A                     CP      '9'+1
02B8' 38 02                     JR      C,prthx2
02BA' C6 07                     ADD     A,7
02BC' CD ** ** prthx2:  CALL    pPRINT
02BF' C9                        RET
02C0'
02C0' F5              pPRTHL:   PUSH    AF
02C1' 7C                        LD      A,H
02C2' CD ** **        CALL      pPRTHX
02C5' 7D                        LD      A,L
02C6' CD ** **        CALL      pPRTHX
02C9' F1                        POP     AF
02CA' C9                        RET
02CB'
02CB'                 ;Print Segment-Mode Into PRN-File
02CB' C5              pPRTSG:   PUSH    BC
02CC' 3A ** **        LD        A,(SEGMD)
02CF' 06 27                     LD      B,"'"
02D1' FE 02                     CP      2       ;CSEG
02D3' 28 08                     JR      Z,pPRTS1
02D5' 06 22                     LD      B,'"'
02D7' FE 03                     CP      3       ;DSEG
02D9' 28 02                     JR      Z,pPRTS1
02DB' 06 21                     LD      B,'!'
02DD' 78              pPRTS1:   LD      A,B
02DE' CD ** **        CALL      pPRINT
02E1' C1                        POP     BC
02E2' C9                        RET
02E3'
02E3'
02E3' F5              pPRNTS:   PUSH    AF
02E4' 3E 20                     LD      A,' '
02E6' CD ** **        CALL      pPRINT
02E9' F1                        POP     AF
02EA' C9                        RET
02EB'
02EB' E5              pPRTXX:   PUSH    HL
02EC' 21 ** **        LD        HL,pass
02EF' 6E                        LD      L,(HL)
02F0' 2D                        DEC     L
02F1' E1                        POP     HL
02F2' C0                        RET     NZ
02F3'
02F3' F5                        PUSH    AF              ;For PASS-2
02F4' C5                        PUSH    BC
02F5' 3E 2A           pRTX1:    LD      A,'*'
02F7' CD ** **        CALL      pPRINT
02FA' 3E 2A                     LD      A,'*'
02FC' CD ** **        CALL      pPRINT
02FF' CD ** **        CALL      pPRNTS
0302' 10 F1                     DJNZ    pRTX1
0304' C1                        POP     BC
0305' F1                        POP     AF
0306' C9                        RET
0307'
0307'                 ;
0307'                 ; Print Text into PRN-file
0307'                 ;
0307' 3A ** ** puttxt:  LD      A,(CURX)
030A' FE 08                     CP      8
030C' DC ** **        CALL      C,putTAB
030F' FE 10                     CP      16
0311' DC ** **        CALL      C,putTAB
0314'
0314' 3A ** **        LD        A,(prname)
0317' A7                        AND     A
0318' 28 23                     JR      Z,PRTT2
031A' 3A ** **        LD        A,(flvl)
031D' A7                        AND     A
031E' 28 1D                     JR      Z,PRTT2
0320'
0320' 3A ** **        LD        A,(CURX)
0323' 47                        LD      B,A
0324' 3E 16                     LD      A,22
0326' 90                        SUB     B
0327' 47                        LD      B,A             ;B = 22 - (CURX)
0328' C5              PRTT1:    PUSH    BC
0329' 3E 20                     LD      A,' '
032B' CD ** **        CALL      putprn
032E' C1                        POP     BC
032F' 10 F7                     DJNZ    PRTT1
0331' 3E 43                     LD      A,'C'
0333' CD ** **        CALL      putprn
0336' 3E 20                     LD      A,' '
0338' CD ** **        CALL      putprn
033B' 18 08                     JR      PRTT3
033D'
033D' 3A ** ** PRTT2:   LD      A,(CURX)
0340' FE 18                     CP      24
0342' DC ** **        CALL      C,putTAB
0345'
0345' 11 ** ** PRTT3:   LD      DE,strbuf
0348' 1A              PRTT4:    LD      A,(DE)
0349' 13                        INC     DE
034A' A7                        AND     A
034B' C8                        RET     Z
034C' CD ** **        CALL      putprn
034F' 18 F7                     JR      PRTT4
0351'
0351' F5              putTAB:   PUSH    AF
0352' 3E 09                     LD      A,TAB
0354' CD ** **        CALL      putprn
0357' 3A ** **        LD        A,(CURX)
035A' C6 08                     ADD     A,8
035C' E6 F8                     AND     0F8H
035E' 32 ** **        LD        (CURX),A
0361' F1                        POP     AF
0362' C9                        RET
0363'
0363'                 ;
0363'                 ; Works
0363'                 ;
0363'                           DSEG
0000"
0000" 00              CURX:     DB      0               ;Cursor position on PRN-buffe
r
0001" 00              TXTFLAG:DB        0               ;Whether Text is to be put
0002"                 PRNBUF:   DS      25
001B"
001B"                           END
```

リスト4 WZDソースリスト3

```
※000'                 ;=================================================
0000'                 ;=                                =
0000'                 ;=      Z80 Relocatable    Assembler =
0000'                 ;=          Programed by T.Ishigami =
0000'                 ;=          Many routins quoted from =
0000'                 ;=            REDA      by T.Takiyama &A.Yuakwa =
0000'                 ;=                                =
0000'                 ;=          '90 May 25th                       =
0000'                 ;=                                =
0000'                 ;=================================================
0000'                 ;
0000'                         CSEG
0000'                 ;
0000'                 ;       Externals
0000'                 ;
                              EXT     PUTrA
                              EXT     PUTrHL
                              EXT     PUTrAHL
                              EXT     PUToA
                              EXT     PUTeA
                              EXT     PUTeHL
                              EXT     PUTeAHL
                              EXT     PUTe82
                              EXT     PUTe84
                              EXT     PUTeF
                              EXT     PUTHD
                              EXT     PUTHD2
                              EXT     PUTHD3
                              EXT     PUTFT
                              EXT     strbuf
                              EXT     prname
                              EXT     include
                              EXT     EVAL
                              EXT     JRVAL
                              EXT     DECI
                              EXT     ERR0
                              EXT     ERR1
                              EXT     ERR2
                              EXT     ERR3
                              EXT     ERR4
                              EXT     ERR5
                              EXT     ERR6
                              EXT     ERR7
                              EXT     ERR8
                              EXT     ERR9
                              EXT     ERR10
                              EXT     ERR11
                              EXT     ERR12
                              EXT     ERR13
                              EXT     ERR14
                              EXT     ERR15
                              EXT     ERR16
                              EXT     POKE_I
                              EXT     PEEK_I
                              EXT     POKE_BC
                              EXT     PEEK_BC
                              EXT     C0TBL
                              EXT     C1TBL
                              EXT     C2TBL
                              EXT     @TBL
                              EXT     REGTBL
                              EXT     RG8TBL
                              EXT     CNDTBL
                              EXT     []TBL
                              EXT     A[]TBL
                              EXT     EXTBL
                              EXT     JPTBL
                              EXT     LSPTBL
                              EXT     SPREND
                              EXT     ADDPC
0000'                 ;
0000'                 ; Characers
0000'                 ;
0009                  TAB       EQU     09H
000D                  CR        EQU     0DH
0020                  SPC       EQU     ' '
0000'                 ;
0000'                 ;         START
0000'                 ;
0000'                           .LIST
0000'                 ;
0000'                 ;         assemble a line
0000'                 ;
0000' ED 73 ** **     linasm::LD        (SPBUF),SP
0004'
0004' 3A ** **        LD        A,(prname)
0007' A7                        AND     A
0008' C4 ** **        CALL      NZ,PUTHD
000B'
000B' CD ** **        CALL      GETADRS
000E' 22 ** **        LD        (PADR),HL
0011'
0011' 2A ** **        LD        HL,(LINNO)
0014' 23                        INC     HL
0015' 22 ** **        LD        (LINNO),HL
0018'
0018' 11 ** **        LD        DE,strbuf
001B' ED 53 ** **               LD      (LINPTR),DE
001F'
001F' CD ** **        CALL      DLABEL          ;Define Label (if exits )
0022' CD ** **        CALL      SPCUT
0025' 1A                        LD      A,(DE)
0026' FE 3B                     CP      ';'
0028' 28 3D                     JR      Z,NXLIN
002A' FE 0D                     CP      CR
002C' 28 39                     JR      Z,NXLIN
002E' FE 2E                     CP      '.'
0030' 28 44                     JR      Z,@COM
0032' D6 41                     SUB     'A'
0034' DA ** **        JP        C,ERR4
0037' FE 1A                     CP      'Z'-'A'+1
0039' D2 ** **        JP        NC,ERR4
003C' 87                        ADD     A,A
003D' 4F                        LD      C,A
003E' 06 00                     LD      B,0
0040' 21 ** **        LD        HL,C0TBL
0043' 09                        ADD     HL,BC
0044' 7E                        LD      A,(HL)
0045' 23                        INC     HL
0046' 66                        LD      H,(HL)
0047' 6F                        LD      L,A
0048' EB              LNASM1:   EX      DE,HL
0049' CD ** **        CALL      SEAOP
004C' EB                        EX      DE,HL
004D' DA ** **        JP        C,ERR4
0050' CD ** **        CALL      SPCUT
0053' 08                        EX      AF,AF'
0054' CD ** **        CALL      GENCOD
0057'
0057' CD ** **        CALL      SPCUT
005A' 1A                        LD      A,(DE)
005B' A7                        AND     A
005C' 28 0F                     JR      Z,LNASM2
005E' FE 0D                     CP      CR
0060' 28 0B                     JR      Z,LNASM2
0062' FE 3B                     CP      ';'
0064' C2 ** **        JP        NZ,ERR0                 ;SyntaxError
0067'
0067' 1A              NXLIN::   LD      A,(DE)
0068' 13                        INC     DE
0069' FE 0D                     CP      CR
006B' 20 FA                     JR      NZ,NXLIN
006D'
006D' 3A ** ** LNASM2:  LD      A,(prname)
0070' A7                        AND     A
0071' C4 ** **        CALL      NZ,PUTFT
0074' B7                        OR      A
0075' C9                        RET
0076'
0076' 21 ** ** @COM:    LD      HL,@TBL
0079' 18 CD                     JR      LNASM1
007B'
007B'                 ;
007B'                 ; Define Labels (called     from linasm)
007B'                 ;
007B' 1A              DLABEL:   LD      A,(DE)
007C' FE 40                     CP      '@'             ;Whther A can be Label 1st
Character
007E' D8                        RET     C
007F' CD ** **        CALL      SKPLBL          ;Skip label strings
0082' CD ** **        CALL      SPCUT
0085' 21 ** **        LD        HL,SDBLK
0088' CD ** **        CALL      amatch
008B' 28 19                     JR      Z,LAB1          ;LABEL:: --
008D' 21 ** **        LD        HL,SEQU
0090' CD ** **        CALL      amatch
0093' 28 63                     JR      Z,LAB2          ;LABELEQU --
0095' 1A                        LD      A,(DE)
0096' 13                        INC     DE
0097' FE 3A                     CP      ':'
0099' CA ** **        JP        Z,LAB3          ;LABEL:  --
009C' C3 ** **        JP        ERR0            ;Syntax Error
009F'
009F' 3A 3A 00 SDBLK:   DB      '::',0
00A2' 45 51 55 00     SEQU:     DB      'EQU',0
00A6'
00A6'                 ;
00A6'                 ; LABEL:: --
00A6'                 ;
00A6' 3A ** ** LAB1:    LD      A,(pass)
00A9' A7                        AND     A
00AA' 28 33                     JR      Z,LLAB3
00AC'
00AC' 3E 87                     LD      A,87H           ;For PASS-2
00AE' CD ** **        CALL      PUTrA
00B1' ED 5B ** **               LD      DE,(LINPTR)
00B5' CD ** **        CALL      SEALBL
00B8' DA ** **        JP        C,ERR1          ;Undefined Label
00BB' 3E 81                     LD      A,81H
00BD' CD ** **        CALL      PUTrAHL
00C0'
00C0' ED 5B ** **               LD      DE,(LINPTR)
00C4' CD ** **        CALL      LENLBL
00C7' 78                        LD      A,B
00C8' FE 20                     CP      31+1
00CA' D2 ** **        JP        NC,ERR3                 ;Too Long Label Name
00CD'
00CD' CD ** **        CALL      PUTrAHL
00D0'
00D0' 1A              LLAB1:    LD      A,(DE)
00D1' CD ** **        CALL      ISALNUM
00D4' 38 06                     JR      C,LLAB2
00D6' 13                        INC     DE
00D7' CD ** **        CALL      PUTrA
00DA' 18 F4                     JR      LLAB1
00DC'
00DC' 13              LLAB2:    INC     DE
00DD' 13                        INC     DE              ;Skip '::'
00DE' C9                        RET
00DF'
00DF' CD ** ** LLAB3:   CALL    GETADRS                 ;For PASS-1
00E2' 44                        LD      B,H
00E3' 4D                        LD      C,L
00E4' CD ** **        CALL      GETSEG
00E7' ED 5B ** **               LD      DE,(LINPTR)
00EB' 2A ** **        LD        HL,(LBLNUM)
00EE' 23                        INC     HL
00EF' 22 ** **        LD        (LBLNUM),HL
00F2' CD ** **        CALL      DEFLBL
```

▶STUDIO Xに呪一平，ゼンマイチャン，K君と書いてあるのを見て弟が「パンクロー，ジョセフソン」と口ずさむ。思い出させるな！　そんなものを！　最近出てこないのが不幸中の幸いだ。

溝渕　誠（19）大阪府

```
00F5' 13                     INC   DE
00F6' 13                     INC   DE          ;Skip '::'
00F7' C9                     RET
00F8'                  ;
00F8'                  ; LABEL EQU ---
00F8'                  ;
00F8' 3A ** ** LAB2:   LD    A,(pass)
00FB' A7                     AND   A
00FC' 20 17                  JR    NZ,LLAB4
00FE'
00FE' CD ** **         CALL  EVAL        ;For Pass-1
0101' 3A ** **         LD    A,(UNDEF)
0104' E6 01                  AND   1
0106' C2 ** **         JP    NZ,ERR15    ;Value Error
0109'
0109' D5                     PUSH  DE
010A' 3E 01                  LD    A,1         ;Enable Only ASEG
010C' ED 5B ** **            LD    DE,(LINPTR)
0110' CD ** **         CALL  DEFLBL      ;  LABEL EQU --
0113' D1                     POP   DE          ;DE points
0114' C9                     RET
0115'
0115' 3A ** ** LLAB4:  LD    A,(prname)
0118' A7                     AND   A
0119' 28 09                  JR    Z,LLAB7
011B'
011B' CD ** **         CALL  EVAL
011E' 60                     LD    H,B
011F' 69                     LD    L,C
0120' CD ** **         CALL  PUTHD2      ;PUT The EQU-Value into PRN-FILE
0123' C9                     RET
0124'
0124' 1A               LLAB7:  LD    A,(DE)
0125' FE 0D                  CP    CR
0127' C8                     RET   Z
0128' 13                     INC   DE
0129' 18 F9                  JR    LLAB7
012B'
012B'                  ;
012B'                  ; LABEL: --
012B'                  ;
012B' 3A ** ** LAB3:   LD    A,(pass)
012E' A7                     AND   A
012F' 20 29                  JR    NZ,LLAB6
0131'
0131' CD ** **         CALL  GETADRS           ;For PASS-1
0134' 44                     LD    B,H
0135' 4D                     LD    C,L
0136' 3A ** **         LD    A,(LOCFLG)
0139' A7                     AND   A
013A' 20 13                  JR    NZ,LLAB5
013C'
013C' ED 5B ** **            LD    DE,(LINPTR)
0140' CD ** **         CALL  GETSEG
0143' 2A ** **         LD    HL,(LBLNUM)
0146' 23                     INC   HL
0147' 22 ** **         LD    (LBLNUM),HL
014A' CD ** **         CALL  DEFLBL
014D' 13                     INC   DE          ;Skip ':'
014E' C9                     RET
014F'
014F' 3E 01            LLAB5:  LD    A,1         ;ASEG
0151' ED 5B ** **            LD    DE,(LINPTR)
0155' CD ** **         CALL  DEFLBL
0158' 13                     INC   DE          ;Skip ':'
0159' C9                     RET
015A'
015A' ED 5B ** **      LLAB6:  LD    DE,(LINPTR) ;For PASS-2
015E' CD ** **         CALL  SEALBL
0161' 13                     INC   DE          ;Skip ':'
0162' FE 01                  CP    1
0164' C8                     RET   Z           ;RET if the label is ASEG
0165'
0165' 3E 87                  LD    A,87H
0167' CD ** **         CALL  PUTrA
016A' 3E 81                  LD    A,81H
016C' CD ** **         CALL  PUTrAHL
016F' C9                     RET
0170'
0170'                  ;
0170'                  ;Define a label
0170'                  ;
0170' 32 ** ** DEFLBL: LD    (SV_ATR),A
0173' ED 43 ** **            LD    (SV_VAR),BC
0177' 22 ** **         LD    (SV_NUM),HL
017A' D5                     PUSH  DE
017B' CD ** **         CALL  SEALBL
017E' D1                     POP   DE
017F' D2 ** **         JP    NC,ERR2           ;Multi defined Error
0182'
0182' D5                     PUSH  DE
0183' CD ** **         CALL  HASH
0186' D1                     POP   DE
0187' 26 00                  LD    H,0
0189' 6F                     LD    L,A
018A' 29                     ADD   HL,HL       ;HL = HASH * 2
018B'
018B' 44                     LD    B,H
018C' 4D                     LD    C,L         ;BC = LABEL BUFFER POINTER
018D'
018D' 60               DEF1:   LD    H,B
018E' 69                     LD    L,C
018F' CD ** **         CALL  PEEK_BC
0192' 78                     LD    A,B
0193' B1                     OR    C
0194' 20 F7                  JR    NZ,DEF1
0196'
0196'                  ; Chain Last struct with New struct
0196' 2B                     DEC   HL
0197' 2B                     DEC   HL
0198' ED 4B ** **            LD    BC,(LBLPTR)
019C' CD ** **         CALL  POKE_BC
019F'
019F'                  ;** Make Struct of Labels **
019F' 2A ** **         LD    HL,(LBLPTR)
01A2' 01 00 00         LD    BC,0
01A5' CD ** **         CALL  POKE_BC
01A8' 3A ** **         LD    A,(SV_ATR)
01AB' CD ** **         CALL  POKE_I
01AE' ED 4B ** **            LD    BC,(SV_NUM)
01B2' CD ** **         CALL  POKE_BC
01B5' ED 4B ** **            LD    BC,(SV_VAR)
01B9' CD ** **         CALL  POKE_BC
01BC'
01BC' 1A               DEF2:   LD    A,(DE)
01BD' CD ** **         CALL  ISALNUM
01C0' 38 06                  JR    C,DEF3
01C2' 13                     INC   DE
01C3' CD ** **         CALL  POKE_I
01C6' 18 F4                  JR    DEF2
01C8'
01C8' AF               DEF3:   XOR   A
01C9' CD ** **         CALL  POKE_I
01CC' 22 ** **         LD    (LBLPTR),HL
01CF' B7                     OR    A           ;CY = 0
01D0' C9                     RET
01D1'
01D1'                  SV_ATR: DS    1
01D2'                  SV_NUM: DS    2
01D4'                  SV_VAR: DS    2
01D6'
01D6'                  ;
01D6'                  ;       Search Label
01D6'                  ;
01D6' ED 53 ** **      SEALBL::LD     (SV_PTR),DE
01DA'
01DA' CD ** **         CALL   HASH
01DD' 26 00                   LD    H,0
01DF' 6F                      LD    L,A
01E0' 29                      ADD   HL,HL
01E1' 22 ** **         LD     (SV_STR),HL
01E4'
01E4' 2A ** ** SEA1:   LD     HL,(SV_STR)
01E7' CD ** **         CALL   PEEK_BC
01EA' 78                      LD    A,B
01EB' B1                      OR    C
01EC' 20 0A                   JR    NZ,SEA4
01EE'
01EE' ED 5B ** **             LD    DE,(SV_PTR)
01F2' CD ** **         CALL   SKPLBL
01F5' AF                      XOR   A           ;Z = 0
01F6' 37                      SCF
01F7' C9                      RET               ;not Defined
01F8'
01F8' ED 43 ** **      SEA4:  LD    (SV_STR),BC
01FC' 21 07 00         LD     HL,7
01FF' 09                      ADD   HL,BC       ;HL = &(Label->str)
0200'
0200'                  ; Check The Labels are the same
0200'
0200' ED 5B ** **             LD    DE,(SV_PTR)
0204' 1A               SEA2:  LD    A,(DE)
```

```
0205' CD ** **         CALL   ISALNUM
0208' 38 0A                   JR    C,SEA3      ;The Same !!
020A' 13                      INC   DE
020B' 47                      LD    B,A
020C' CD ** **         CALL   PEEK_I
020F' B8                      CP    B
0210' 28 F2                   JR    Z,SEA2
0212'
0212' 18 D0                   JR    SEA1        ;Not The Same
0214'
0214' CD 94 1F SEA3:   CALL   _PEEK
0217' A7                      AND   A           ;END CODE ?
0218' 20 CA                   JR    NZ,SEA1     ;Not The Same
021A'
021A' 2A ** **         LD     HL,(SV_STR)
021D' 23                      INC   HL
021E' 23                      INC   HL
021F' CD ** **         CALL   PEEK_I      ;A = Attribute of the label
0222' CD ** **         CALL   PEEK_BC
0225' C5                      PUSH  BC
0226' CD ** **         CALL   PEEK_BC           ;BC= the value
0229' E1                      POP   HL          ;HL= label number
022A' B7                      OR    A           ;CY = 0
022B' C9                      RET
022C'
022C'                  SV_STR: DS    2
022E'                  SV_PTR: DS    2
0230'                  ;
0230'                  ; Counts Characters of labels
0230'                  ;
0230' D5               LENLBL::PUSH  DE
0231' 06 FF                   LD    B,-1
0233' 1A               LENL1: LD    A,(DE)
0234' 13                      INC   DE
0235' 04                      INC   B
0236' CD ** **         CALL   ISALNUM
0239' 30 F8                   JR    NC,LENL1
023B' D1                      POP   DE
023C' C9                      RET
023D'                  ;
023D'                  ; Compare Strings pointed HL & DE
023D'                  ;       if match proceed DE
023D' D5               amatch: PUSH  DE
023E' 7E               amat1: LD    A,(HL)
023F' A7                      AND   A
0240' 28 08                   JR    Z,amat2
0242' 1A                      LD    A,(DE)
0243' 13                      INC   DE
0244' BE                      CP    (HL)
0245' 23                      INC   HL
0246' 28 F6                   JR    Z,amat1
0248' D1                      POP   DE
0249' C9                      RET         ;with Z = 0 (Not matched)
024A'
024A' E1               amat2: POP   HL    ;Give up old DE
024B' C9                      RET         ;with Z = 1 (matched)
024C'
024C'                  ;
024C'                  ; Skip Label strings
024C'                  ;
024C' 1A               SKPLBL::LD    A,(DE)
024D' CD ** **         CALL   ISALNUM
0250' D8                      RET   C
0251' 13                      INC   DE
0252' 18 F8                   JR    SKPLBL
0254'                  ;
0254'                  ; Is the character in Areg. alphabet or deicmal ?
0254'                  ;
0254'                  ISALNUM::
0254' FE 40                   CP    '@'
0256' D0                      RET   NC
0257' FE 3A            ISDEC:: CP    '9'+1
0259' 3F                      CCF
025A' D8                      RET   C
025B' FE 30                   CP    '0'
025D' C9                      RET
025E'
025E'                  ;
025E'                  ;       calculate hash of label
025E'                  ;
025E' C5               HASH:  PUSH  BC
025F' 06 00                   LD    B,0
0261' 1A               HASH0: LD    A,(DE)
0262' CD ** **         CALL   ISALNUM
0265' 38 05                   JR    C,HASH1
0267' 13                      INC   DE
0268' 80                      ADD   A,B
0269' 47                      LD    B,A
026A' 18 F5                   JR    HASH0
026C' 78               HASH1: LD    A,B
026D' C1                      POP   BC
026E' C9                      RET
026F'                  ;
026F'                  ; GET Segment Mode
026F'                  ;
026F' 3A ** ** GETSEG: LD     A,(LOCFLG)
0272' A7                      AND   A
0273' 3E 01                   LD    A,1         ;ASEG
0275' C0                      RET   NZ
0276' 3A ** **         LD     A,(SEGMD)
0279' C9                      RET
027A'
027A'                  ;
027A'                  ; GET ADDRESS
027A'                  ;
027A'                  GETADRS::
027A' 2A ** **         LD     HL,(PTRFC)
027D' 3A ** **         LD     A,(LOCFLG)
0280' A7                      AND   A
0281' C0                      RET   NZ
0282' 2A ** **         LD     HL,(PTRCD)
0285' 3A ** **         LD     A,(SEGMD)
0288' FE 02                   CP    2
028A' C8                      RET   Z
028B' 2A ** **         LD     HL,(PTRDT)
028E' FE 03                   CP    3
0290' C8                      RET   Z
0291' 2A ** **         LD     HL,(PTRWK)
0294' C9                      RET
0295'
0295'                  ;
0295'                  ;       search opecode
0295'                  ;
0295' 23               SEAOP: INC   HL
0296' 4D                      LD    C,L
0297' 44                      LD    B,H
0298' 1A               SEAOP0: LD    A,(DE)
0299' B7                      OR    A
029A' 37                      SCF
029B' C8                      RET   Z           ;CY=1
029C' 1A               SEAOP1: LD    A,(DE)
029D' 13                      INC   DE
029E' B7                      OR    A
029F' FA ** **         JP     M,SEAOP2
02A2' BE                      CP    (HL)
02A3' 20 0C                   JR    NZ,SEAOP3
02A5' 23                      INC   HL
02A6' C3 ** **         JP     SEAOP1
02A9' 08               SEAOP2: EX    AF,AF'
02AA' 7E                      LD    A,(HL)
02AB' CD ** **         CALL   ISBLNK
02AE' 20 07                   JR    NZ,SEAOP4
02B0' C9                      RET               ;NC,AF'=no
02B1' 1A               SEAOP3: LD    A,(DE)
02B2' 13                      INC   DE
02B3' B7                      OR    A
02B4' F2 ** **         JP     P,SEAOP3
02B7' 69               SEAOP4: LD    L,C
02B8' 60                      LD    H,B
02B9' C3 ** **         JP     SEAOP0
02BC'                  ;
02BC'                  ;       generate object code
02BC'                  ;
02BC' E6 7F            GENCOD: AND   7FH
02BE' FE 10                   CP    10H
02C0' 38 11                   JR    C,GEN1
02C2' D6 30                   SUB   30H
02C4' 38 3F                   JR    C,GEN2
02C6' 01 ** **         LD     BC,C3TBL
02C9' 6F               TBLJMP: LD    L,A
02CA' 26 00                   LD    H,0
02CC' 29                      ADD   HL,HL
02CD' 09                      ADD   HL,BC
02CE' 4E                      LD    C,(HL)
02CF' 23                      INC   HL
02D0' 66                      LD    H,(HL)
02D1' 69                      LD    L,C
02D2' E9                      JP    (HL)
02D3'                  ;
02D3'                  ;       1byte code
02D3'                  ;
02D3' 4F               GEN1:  LD    C,A
```

```
02D4' 06 00                   LD    B,0
02D6' 21 ** **         LD     HL,C1TBL
02D9' 09                      ADD   HL,BC
02DA' 7E                      LD    A,(HL)
02DB' CD ** **         CALL   PUToA
02DE' C9                      RET
02DF'
02DF' 3E CB            PUTCB: LD    A,0CBH
02E1' C3 ** **         JP     PUToA
02E4' 3E ED            PUTED: LD    A,0EDH
02E6' C3 ** **         JP     PUToA
02E9' 79               PUToBC: LD    A,C
02EA' CD ** **         CALL   PUToA
02ED' 78                      LD    A,B
02EE' C3 ** **         JP     PUToA
02F1' 79               PUToC: LD    A,C
02F2' C3 ** **         JP     PUToA
02F5'
02F5'                  ;
02F5' FE 0F            PUT[X]: CP    15
02F7' 18 02                   JR    PUTIX0
02F9' FE 0D            PUTIX: CP    13
02FB' 3E DD            PUTIX0: LD    A,0DDH
02FD' CA ** **         JP     Z,PUToA
0300' 3E FD                   LD    A,0FDH
0302' C3 ** **         JP     PUToA
0305'                  ;
0305'                  ;       2bytes code
0305'                  ;
0305' C6 20            GEN2:  ADD   A,20H
0307' 87                      ADD   A,A
0308' 4F                      LD    C,A
0309' 06 00                   LD    B,0
030B' 21 ** **         LD     HL,C2TBL
030E' 09                      ADD   HL,BC
030F' 7E                      LD    A,(HL)
0310' 23                      INC   HL
0311' CD ** **         CALL   PUToA
0314' 7E                      LD    A,(HL)
0315' CD ** **         CALL   PUToA
0318' C9                      RET
0319'                  ;
0319' CD ** ** PUT3:   CALL   PUToA
031C' CD ** **         CALL   PUTe82
031F' C9                      RET
0320'
0320'                  ;
0320' ** ** ** **      C3TBL: DEFW  _ADD, _AND  ;30 31
0324' ** ** ** **             DEFW  _ADC, _BIT  ;32 33
0328' ** ** ** **             DEFW  _CP,  _CALL ;34 35
032C' ** ** ** **             DEFW  _DEC, _DJNZ ;36 37
0330' ** ** ** **             DEFW  _DEFB, _DEFW ;38 39
0334' ** ** ** **             DEFW  _DEFS, _DM  ;3A 3B
0338' ** ** ** **             DEFW  _EX,  ERR0  ;3C 3D;_EX,_EQU
033C' ** ** ** **             DEFW  _INC, _IN   ;3E 3F
0340' ** ** ** **             DEFW  _IM,  _JR   ;40 41
0344' ** ** ** **             DEFW  _JP,  _LD   ;42 43
0348' ** ** ** **             DEFW  _OR,  _OUT  ;44 45
034C' ** ** ** **             DEFW  _ORG, ERR0  ;46 47;_ORG,_OFFSET
0350' ** ** ** **             DEFW  _POP, _PUSH ;48 49
0354' ** ** ** **             DEFW  _RET, _RL   ;4A 4B
0358' ** ** ** **             DEFW  _RR,  _RLC  ;4C 4D
035C' ** ** ** **             DEFW  _RRC, _RES  ;4E 4F
0360' ** ** ** **             DEFW  _RST, _SBC  ;50 51
0364' ** ** ** **             DEFW  _SUB, _SET  ;52 53
0368' ** ** ** **             DEFW  _SLA, _SRA  ;54 55
036C' ** ** ** **             DEFW  _SRL, _XOR  ;56 57
0370' ** ** ** **             DEFW  _LIST, _NLIST ;58 59
0374' ** ** ** **             DEFW  _CSEG, _DSEG ;5A 5B
0378' ** ** ** **             DEFW  _COMMON,_PHASE;5C 5D
037C' ** ** ** **             DEFW  _DEPHASE,_EXT;5E 5F
0380' ** ** ** **             DEFW  _PUBLIC,include;60 61
0384' ** ** ** **             DEFW  _END, _PAGE ;62 63
0388' ** ** ** **             DEFW  _TITLE,     _KANJI;64 65
038C' ** **                   DEFW  _NKANJI     ;66
038E'                  ;
038E' ** ** ** **      LDTBL: DEFW  LD16,LD16
0392' ** ** ** **             DEFW  LDHL,LDSP
0396' ** ** ** **             DEFW  ERR5,LD8
039A' ** ** ** **             DEFW  LD8,LD8
039E' ** ** ** **             DEFW  LD8,LD8
03A2' ** ** ** **             DEFW  LD8,LD8
03A6' ** ** ** **             DEFW  LDA,LDIX
03AA' ** ** ** **             DEFW  LDIX,LD[X]
03AE' ** **                   DEFW  LD[X]
03B0'                  ;
03B0' ** ** ** **      DECTBL: DEFW  DEC16,DEC16
03B4' ** ** ** **             DEFW  DEC16,DEC16
03B8' ** ** ** **             DEFW  ERR5,DEC8
03BC' ** ** ** **             DEFW  DEC8,DEC8
03C0' ** ** ** **             DEFW  DEC8,DEC8
03C4' ** ** ** **             DEFW  DEC8,DEC8
03C8' ** ** ** **             DEFW  DEC8,DECIX
03CC' ** ** ** **             DEFW  DECIX,DEC[X]
03D0' ** **                   DEFW  DEC[X]
03D2'                  ;
03D2' ** ** ** **      INCTBL: DEFW  INC16,INC16
03D6' ** ** ** **             DEFW  INC16,INC16
03DA' ** ** ** **             DEFW  ERR5,INC8
03DE' ** ** ** **             DEFW  INC8,INC8
03E2' ** ** ** **             DEFW  INC8,INC8
03E6' ** ** ** **             DEFW  INC8,INC8
03EA' ** ** ** **             DEFW  INC8,INCIX
03EE' ** ** ** **             DEFW  INCIX,INC[X]
03F2' ** **                   DEFW  INC[X]
03F4'                  ;
03F4' CD ** ** _LD:    CALL   GETREG
03F7' 38 0B                   JR    C,LD[]
03F9' FE 0F                   CP    15
03FB' DC ** **         CALL   C,CMACHK
03FE' 01 ** **         LD     BC,LDTBL
0401' C3 ** **         JP     TBLJMP
0404'                  ;
0404' 21 ** ** LD[]:   LD     HL,[]TBL
0407' CD ** **         CALL   CMPGEN
040A' D0                      RET   NC
040B' CD ** **         CALL   [EVAL]
040E' C5                      PUSH  BC
040F' CD ** **         CALL   CMACHK
0412' CD ** **         CALL   REGCHK
0415' FE 0C                   CP    12          ;A?
0417' CA ** **         JP     Z,LD[]A
041A' FE 04                   CP    4
041C' 38 19                   JR    C,LD[]16
041E' FE 0D                   CP    13
0420' DA ** **         JP     C,ERR5
0423' FE 0F                   CP    15
0425' D2 ** **         JP     NC,ERR5
0428' CD ** ** LD[]IX: CALL   PUTIX
042B' C1               LD[]HL: POP   BC
042C' 3E 22                   LD    A,22H       ;00 100010
042E' C3 ** **         JP     PUT3
0431'                  ;
0431' 3E 32            LD[]A: LD    A,32H       ;00 110010
0433' C1                      POP   BC
0434' C3 ** **         JP     PUT3
0437'                  ;
0437' FE 02            LD[]16: CP    2           ;HL?
0439' 28 F0                   JR    Z,LD[]HL
043B' 87                      ADD   A,A
043C' 87                      ADD   A,A
043D' 87                      ADD   A,A
043E' 87                      ADD   A,A
043F' F6 43                   OR    43H         ;01 XX0 011
0441' F5                      PUSH  AF
0442' CD ** **         CALL   PUTED
0445' F1                      POP   AF
0446' C1                      POP   BC
0447' C3 ** **         JP     PUT3
044A'                  ;
044A' CD ** ** LDSP:   CALL   GETREG
044D' 38 17                   JR    C,LDSP0
044F' 0E F9                   LD    C,0F9H
0451' FE 02                   CP    2           ;HL?
0453' CA ** **         JP     Z,PUToC
0456' FE 0D                   CP    13          ;IX?
0458' 01 DD F9         LD     BC,0F9DDH
045B' CA ** **         JP     Z,PUToBC
045E' FE 0E                   CP    14          ;IY?
0460' 01 FD F9         LD     BC,0F9FDH
0463' CA ** **         JP     Z,PUToBC
0466' 3E 03            LDSP0: LD    A,3
0468'                         ;
0468' 87               LD16:  ADD   A,A
0469' 87                      ADD   A,A
046A' 87                      ADD   A,A
046B' 87                      ADD   A,A
046C' F5                      PUSH  AF
046D' CD ** **         CALL   IS[
0470' 28 0C                   JR    Z,LD[16]
0472' F1                      POP   AF
0473' F6 01                   OR    01H         ;00 XX0 001
```

▶"この木なんの木" で山瀬まみを思い出したのは，私だけだろうか？
高垣　卓也（26）大阪府

```
0475' CD ** **          CALL    PUToA
0478' CD ** **          CALL    EVAL
047B' C3 ** **          JP      PUTe82
047E'                   ;
047E' CD ** ** LD[16]:  CALL    PUTED
0481' F1                        POP     AF
0482' F6 4B                     OR      4BH             ;01 XX1011
0484' CD ** **          CALL    PUToA
0487' CD ** **          CALL    EVAL]
048A' C3 ** **          JP      PUTe82
048D'                   ;
048D' CD ** ** LDIX:    CALL    PUTIX
0490' CD ** ** LDHL:    CALL    IS[
0493' 28 0B                     JR      Z,LDHL[]
0495' 3E 21                     LD      A,21H           ;LD HL,nn
0497' CD ** **          CALL    PUToA
049A' CD ** **          CALL    EVAL
049D' C3 ** **          JP      PUTe82
04A0' 3E 2A             LDHL[]: LD      A,2AH           ;LD HL,(nn)
04A2' CD ** **          CALL    PUToA
04A5' CD ** **          CALL    EVAL]
04A8' C3 ** **          JP      PUTe82
04AB'                   ;
04AB' D6 05             LDA8:   SUB     5
04AD' F6 78                     OR      78H             ;01 111 000
04AF' C3 ** **          JP      PUToA
04B2'                   ;
04B2' CD ** ** LDA:     CALL    GETREG
04B5' 38 12                     JR      C,LDA0
04B7' FE 05                     CP      5               ;rp?
04B9' 38 5C                     JR      C,ER5C0
04BB' FE 0D                     CP      13              ;reg?
04BD' 38 EC                     JR      C,LDA8
04BF' FE 0F                     CP      15              ;index reg?
04C1' 38 54                     JR      C,ER5C0
04C3' 06 38             LDA[X]: LD      B,38H
04C5' C5                        PUSH    BC
04C6' C3 ** **          JP      LD8[X]0
04C9'                   ;
04C9' CD ** ** LDA0:    CALL    IS[
04CC' 28 2A                     JR      Z,LDA[]
04CE' FE 49                     CP      'I'
04D0' 28 1A                     JR      Z,LDAI
04D2' FE 52                     CP      'R'
04D4' 20 0B                     JR      NZ,LDAN
04D6' 13                LDAR:   INC     DE
04D7' CD ** **          CALL    ISBLK0
04DA' 01 ED 5F          LD      BC,05FEDH     ;LD A,R
04DD' CA ** **          JP      Z,PUToBC
04E0' 1B                LDAR0:  DEC     DE
04E1' CD ** ** LDAN:    CALL    EVAL
04E4' 3E 3E                     LD      A,3EH           ;LD A,n
04E6' CD ** **          CALL    PUToA
04E9' C3 ** **          JP      PUTe84
04EC' 13                LDAI:   INC     DE
04ED' CD ** **          CALL    ISBLK0
04F0' 20 EE                     JR      NZ,LDAR0
04F2' 01 ED 57          LD      BC,57EDH      ;LD A,I
04F5' C3 ** **          JP      PUToBC
04F8'                   ;
04F8' 21 ** ** LDA[]:   LD      HL,A[]TBL
04FB' CD ** **          CALL    CMPGEN
04FE' D0                        RET     NC
04FF' CD ** **          CALL    EVAL]
0502' 3E 3A                     LD      A,3AH           ;LD A,(nn)
0504' CD ** **          CALL    PUToA
0507' C3 ** **          JP      PUTe82
050A'                   ;
050A' D6 05             LD8:    SUB     5
050C' 87                        ADD     A,A
050D' 87                        ADD     A,A
050E' 87                        ADD     A,A
050F' F5                        PUSH    AF
0510' CD ** **          CALL    GETREG
0513' 38 17                     JR      C,LD8N
0515' FE 05                     CP      5               ;rp?
0517' DA ** ** ER5C0:   JP      C,ERR5
051A' FE 0D                     CP      13              ;index?
051C' 30 1A                     JR      NC,LD8[X]
051E' D6 05                     SUB     5
0520' C1                        POP     BC
0521' B0                        OR      B
0522' F6 40                     OR      40H             ;01 XXX XXX
0524' FE 76                     CP      76H             ;01 110 110
0526' C2 ** **          JP      NZ,PUToA
0529' C3 ** **          JP      ERR5
052C' CD ** ** LD8N:    CALL    EVAL
052F' F1                        POP     AF
0530' F6 06                     OR      06H             ;00 XXX 110
0532' CD ** **          CALL    PUToA
0535' C3 ** **          JP      PUTe84
0538'                   ;
0538' FE 0F             LD8[X]: CP      15              ;index reg?
053A' DA ** **          JP      C,ERR5
053D' CD ** ** LD8[X]0:CALL     PUT[X]D
0540' F1                        POP     AF
0541' FE 30                     CP      30H             ;XXX LD (HL),(IX+d)
0543' CA ** **          JP      Z,ERR5
0546' F6 46                     OR      46H             ;01 000 110
0548' CD ** **          CALL    PUToA
054B' C3 ** **          JP      PUTe84
054E'                   ;
054E' CD ** ** LD[X]:   CALL    PUT[X]D
0551' CD ** **          CALL    CMACHK
0554' CD ** **          CALL    GETRG8
0557' 38 08                     JR      C,LD[X]N
0559' F6 70                     OR      70H             ;01 110 XXX
055B' CD ** **          CALL    PUToA
055E' C3 ** **          JP      PUTe84
0561'                   ;
0561' 3E 36             LD[X]N: LD      A,36H           ;LD (HL),n
0563' CD ** **          CALL    PUToA
0566' CD ** **          CALL    PUTe84
0569' CD ** **          CALL    EVAL
056C' C3 ** **          JP      PUTe84
056F'                   ;
056F' 21 ** ** _EX:     LD      HL,EXTBL
0572' CD ** **          CALL    CMPGEN
0575' D0                        RET     NC
0576' C3 ** ** ERR51:   JP      ERR5
0579'                   ;
0579' 0E 88             ADCA:   LD      C,88H           ;10 001 000
057B' 21                        DEFB    21H
057C' 0E 98             SBCA:   LD      C,98H           ;10 011 000
057E' 21                        DEFB    21H
057F' 0E A0             _AND:   LD      C,0A0H          ;10 100 000
0581' 21                        DEFB    21H
0582' 0E A8             _XOR:   LD      C,0A8H          ;10 101 000
0584' 21                        DEFB    21H
0585' 0E 90             _SUB:   LD      C,90H           ;10 010 000
0587' 21                        DEFB    21H
0588' 0E 80             ADDA:   LD      C,80H           ;10 000 000
058A' 21                        DEFB    21H
058B' 0E B0             _OR:    LD      C,0B0H          ;10 110 000
058D' 21                        DEFB    21H
058E' 0E B8             _CP:    LD      C,0B8H          ;10 111 000
0590'                   ;
0590' CD ** **          CALL    GETREG
0593' 38 10                     JR      C,CPN
0595' FE 0F                     CP      15              ;(IX+d)?
0597' 30 18                     JR      NC,CP[X]
0599' D6 05                     SUB     5
059B' 38 D9                     JR      C,ERR51         ;rp
059D' FE 08                     CP      8
059F' 30 D5                     JR      NC,ERR51        ;index reg
05A1' B1                        OR      C
05A2' C3 ** **          JP      PUToA
05A5' 79                CPN:    LD      A,C
05A6' F6 C6                     OR      0C6H            ;11 000 110
05A8' CD ** **          CALL    PUToA
05AB' CD ** **          CALL    EVAL
05AE' C3 ** **          JP      PUTe84
05B1'                   ;
05B1' CD ** ** CP[X]:   CALL    PUT[X]
05B4' 3E 86                     LD      A,086H          ;10 000 110
05B6' B1                        OR      C
05B7' CD ** **          CALL    PUToA
05BA' CD ** **          CALL    GETIXD
05BD' C3 ** **          JP      PUToC
05C0'                   ;
05C0' CD ** ** _ADD:    CALL    REGCHK
05C3' CD ** **          CALL    CMACHK
05C6' FE 0C                     CP      12              ;A?
05C8' 28 BE                     JR      Z,ADDA
05CA' FE 02                     CP      2               ;HL?
05CC' 28 17                     JR      Z,ADDHL
05CE' FE 0D                     CP      13              ;index reg?
05D0' 38 A4                     JR      C,ERR51
05D2' FE 0F                     CP      15              ;(IX+d)?
05D4' 30 A0                     JR      NC,ERR51
05D6' F5                ADDIX:  PUSH    AF
05D7' CD ** **          CALL    PUTIX
05DA' CD ** **          CALL    REGCHK
05DD' C1                        POP     BC
05DE' B8                        CP      B
05DF' 20 07                     JR      NZ,ADDHL0
05E1' 3E 02                     LD      A,02H           ;HL
05E3' 18 07                     JR      ADDHL1
05E5' CD ** ** ADDHL:   CALL    REGCHK
05E8' FE 04             ADDHL0: CP      4               ;rp?
05EA' 30 8A                     JR      NC,ERR51        ;
05EC' 0E 09             ADDHL1: LD      C,09H           ;00 XX1 001
05EE' 18 25                     JR      PUSH1
05F0'                   ;
05F0' 0E C5             _PUSH:  LD      C,0C5H          ;11 XX0 101
05F2' 21                        DEFB    21H
05F3' 0E C1             _POP:   LD      C,0C1H          ;11 XX0 001
05F5' CD ** **          CALL    REGCHK
05F8' FE 03                     CP      3               ;SP?
05FA' CA ** **          JP      Z,ERR5        ;XXX POP SP
05FD' 38 16                     JR      C,PUSH1
05FF' FE 04                     CP      4               ;AF?
0601' 28 11                     JR      Z,PUSH0
0603' FE 0D                     CP      13              ;reg?
0605' DA ** **          JP      C,ERR5
0608' FE 0F                     CP      15              ;index reg?
060A' D2 ** **          JP      NC,ERR5
060D' CD ** **          CALL    PUTIX
0610' 3E 02                     LD      A,02H           ;HL
0612' 18 01                     JR      PUSH1
0614' 3D                PUSH0:  DEC     A               ;SP
0615' 87                PUSH1:  ADD     A,A
0616' 87                PUSH2:  ADD     A,A
0617' 87                        ADD     A,A
0618' 87                        ADD     A,A
0619' B1                        OR      C
061A' C3 ** **          JP      PUToA
061D'                   ;
061D' CD ** ** _SBC:    CALL    REGCHK
0620' CD ** **          CALL    CMACHK
0623' FE 0C                     CP      12              ;A?
0625' CA ** **          JP      Z,SBCA
0628' 0E 42                     LD      C,42H           ;01 XX0 010
062A' 18 0D                     JR      ADC0
062C'                   ;
062C' CD ** ** _ADC:    CALL    REGCHK
062F' CD ** **          CALL    CMACHK
0632' FE 0C                     CP      12              ;A?
0634' CA ** **          JP      Z,ADCA
0637' 0E 4A                     LD      C,4AH           ;01 XX1 010
0639' FE 02             ADC0:   CP      2               ;HL?
063B' C2 ** **          JP      NZ,ERR5         ;
063E' CD ** **          CALL    PUTED
0641' CD ** **          CALL    REGCHK
0644' FE 04                     CP      4               ;rp?
0646' 38 CD                     JR      C,PUSH1
0648' C3 ** **          JP      ERR5
064B'                   ;
064B' CD ** ** _DEC:    CALL    REGCHK
064E' 01 ** **          LD      BC,DECTBL
0651' C3 ** **          JP      TBLJMP
0654'                   ;
0654' D6 05             DEC8:   SUB     5
0656' 0E 05                     LD      C,05H           ;00 XXX 101
0658' C3 ** **          JP      PUSH2
065B'                   ;
065B' 0E 0B             DEC16:  LD      C,0BH           ;00 XX1 011
065D' C3 ** **          JP      PUSH1
0660'                   ;
0660' CD ** ** DECIX:   CALL    PUTIX
0663' 3E 2B                     LD      A,2BH           ;00 101 011
0665' C3 ** **          JP      PUToA
0668'                   ;
0668' CD ** ** DEC[X]:  CALL    PUT[X]D
066B' 3E 35                     LD      A,35H           ;00 110 101
066D' CD ** **          CALL    PUToA
0670' 3E 84                     LD      A,84H
0672' C3 ** **          JP      PUTrA
0675'                   ;
0675' CD ** ** _INC:    CALL    REGCHK
0678' 01 ** **          LD      BC,INCTBL
067B' C3 ** **          JP      TBLJMP
067E'                   ;
067E' D6 05             INC8:   SUB     5
0680' 0E 04                     LD      C,04H           ;00 XXX 100
0682' C3 ** **          JP      PUSH2
0685'                   ;
0685' 0E 03             INC16:  LD      C,03H           ;00 XX0 011
0687' C3 ** **          JP      PUSH1
068A'                   ;
068A' CD ** ** INCIX:   CALL    PUTIX
068D' 3E 23                     LD      A,23H           ;00 100 011
068F' C3 ** **          JP      PUToA
0692'                   ;
0692' CD ** ** INC[X]:  CALL    PUT[X]D
0695' 3E 34                     LD      A,34H           ;00 110 100
0697' CD ** **          CALL    PUToA
069A' 3E 84                     LD      A,84H
069C' C3 ** **          JP      PUTrA
069F'                   ;
069F' 0E 10             _DJNZ:  LD      C,10H           ;DJNZ d
06A1' 18 15                     JR      JR0
06A3'                   ;
06A3' CD ** ** _JR:     CALL    GETCND
06A6' 0E 18                     LD      C,18H           ;JR d
06A8' 38 0E                     JR      C,JR0
06AA' FE 04                     CP      4
06AC' D2 ** **          JP      NC,ERR5         ;XXX PO,PE,P,M
06AF' CD ** **          CALL    CMACHK
06B2' 87                        ADD     A,A
06B3' 87                        ADD     A,A
06B4' 87                        ADD     A,A
06B5' F6 20                     OR      20H             ;00 1XX000
06B7' 4F                        LD      C,A
06B8' 79                JR0:    LD      A,C
06B9' CD ** **          CALL    PUToA
06BC'
06BC' CD ** **          CALL    JRVAL
06BF'
06BF' 3A ** **          LD      A,(pass)
06C2' A7                        AND     A
06C3' CA ** **          JP      Z,PUToA         ;PUT Dummy Data ,when PASS-1
06C6'
06C6' 3A ** **          LD      A,(LOCFLG)
06C9' A7                        AND     A
06CA' 28 0A                     JR      Z,JR1
06CC' 3A ** **          LD      A,(JRSEG)
06CF' FE 01                     CP      1
06D1' C2 ** **          JP      NZ,ERR16      ;Relocation Err
06D4' 18 0B                     JR      JR2
06D6'
06D6' 3A ** ** JR1:     LD      A,(JRSEG)
06D9' 47                        LD      B,A
06DA' 3A ** **          LD      A,(SEGMD)
06DD' B8                        CP      B
06DE' C2 ** **          JP      NZ,ERR16      ;Relocation Err
06E1'
06E1' E5                JR2:    PUSH    HL
06E2' CD ** **          CALL    GETADRS
06E5' 44                        LD      B,H
06E6' 4D                        LD      C,L
06E7' E1                        POP     HL
06E8' 37                        SCF
06E9' ED 42                     SBC     HL,BC           ;HL = HL - (BC+1)
06EB' 38 0D                     JR      C,JR3
06ED' 25                        DEC     H
06EE' 24                        INC     H
06EF' C2 ** **          JP      NZ,ERR8         ;Too Far
06F2' 7D                        LD      A,L
06F3' B7                        OR      A
06F4' F2 ** **          JP      P,PUToA
06F7' C3 ** **          JP      ERR8          ;Too Far
06FA' 24                JR3:    INC     H
06FB' C2 ** **          JP      NZ,ERR8
06FE' 7D                        LD      A,L
06FF' B7                        OR      A
0700' FA ** **          JP      M,PUToA
0703' C3 ** **          JP      ERR8
0706'
0706'
0706' CD ** ** _JP:     CALL    GETCND
0709' 0E C2                     LD      C,0C2H          ;11 XXX 010
070B' 30 12                     JR      NC,CALL1
070D' 21 ** **          LD      HL,JPTBL
0710' CD ** **          CALL    CMPGEN
0713' D0                        RET     NC
0714' 3E C3                     LD      A,0C3H          ;JP nn
0716' 18 11                     JR      CALL3
0718'
0718' CD ** ** _CALL:   CALL    GETCND
071B' 38 0A                     JR      C,CALL2
071D' 0E C4             CALL0:  LD      C,0C4H          ;11 XXX 100
071F' CD ** ** CALL1:   CALL    PUSH2
0722' CD ** **          CALL    CMACHK
0725' 18 05                     JR      CALL4
0727' 3E CD             CALL2:  LD      A,0CDH          ;CALL nn
0729' CD ** ** CALL3:   CALL    PUToA
072C' CD ** ** CALL4:   CALL    EVAL
072F' CD ** **          CALL    PUTe82
0732' C9                        RET
0733'                   ;
0733' CD ** ** _RET:    CALL    ISBLK0
0736' 3E C9                     LD      A,0C9H          ;RET
0738' CA ** **          JP      Z,PUToA
073B' CD ** **          CALL    GETCND
073E' DA ** **          JP      C,ERR5
0741' 0E C0                     LD      C,0C0H          ;11 XXX 000
0743' C3 ** **          JP      PUSH2
0746'                   ;
0746' CD ** ** _IN:     CALL    GETREG
0749' DA ** **          JP      C,ERR5
074C' CD ** **          CALL    CMACHK
074F' D6 05                     SUB     5
0751' DA ** **          JP      C,ERR5        ;rp
0754' FE 06                     CP      6               ;(HL)?
0756' CA ** **          JP      Z,ERR5
0759' FE 07                     CP      7               ;A?
075B' 28 14                     JR      Z,INA
075D' D2 ** **          JP      NC,ERR5
0760' 47                INR:    LD      B,A
0761' CD ** **          CALL    PUTED
0764' 78                        LD      A,B
0765' 0E 40                     LD      C,40H           ;01 XXX 000
0767' CD ** **          CALL    PUSH2
076A' CD ** **          CALL    IS[C]
076D' C8                        RET     Z
076E' C3 ** **          JP      ERR5
0771'                   ;
0771' 21 ** ** INA:     LD      HL,[C]
0774' CD ** **          CALL    CMPGEN
0777' D0                        RET     NC
0778' 3E DB                     LD      A,0DBH          ;IN A,(n)
077A' CD ** **          CALL    PUToA
077D' CD ** **          CALL    [EVAL]
0780' C3 ** **          JP      PUTe84
0783'                   ;
0783' 21 ** ** IS[C]:   LD      HL,[C]
0786' 06 03                     LD      B,3
0788' 1A                IS[C]0: LD      A,(DE)
0789' BE                        CP      (HL)
078A' C0                        RET     NZ
078B' 13                        INC     DE
078C' 23                        INC     HL
078D' 10 F9                     DJNZ    IS[C]0
078F' C9                        RET
0790'                   ;
0790' 28 43 29 00       [C]:    DEFB    '(C)',0
0794' ED 78                     IN      A,(C)
0796' 00                        DEFB    0
0797'                   ;
0797' D5                _OUT:   PUSH    DE
0798' CD ** **          CALL    IS[C]
079B' 28 1A                     JR      Z,OUT[C]
079D' D1                        POP     DE
079E' 3E D3                     LD      A,0D3H          ;OUT (n),A
07A0' CD ** **          CALL    PUToA
07A3' CD ** **          CALL    [EVAL]
07A6' CD ** **          CALL    CMACHK
07A9' CD ** **          CALL    GETRG8
07AC' DA ** **          JP      C,ERR5
07AF' FE 07                     CP      7               ;A?
07B1' C2 ** **          JP      NZ,ERR5
07B4' C3 ** **          JP      PUTe84
07B7'                   ;
07B7' C1                OUT[C]: POP     BC
07B8' CD ** **          CALL    PUTED
07BB' CD ** **          CALL    CMACHK
07BE' CD ** **          CALL    GETRG8
07C1' DA ** **          JP      C,ERR5
07C4' FE 06                     CP      6               ;B-L?
07C6' 0E 41                     LD      C,41H           ;01 XXX  001
07C8' C2 ** **          JP      NZ,PUSH2      ;A
07CB' C3 ** **          JP      ERR5          ;XXX OUT (C),(HL)
07CE'                   ;
07CE' 0E 80             _RES:   LD      C,80H           ;10 XXX XXX
07D0' 21                        DEFB    21H
07D1' 0E C0             _SET:   LD      C,0C0H          ;11 XXX XXX
07D3' 21                        DEFB    21H
07D4' 0E 40             _BIT:   LD      C,40H           ;01 XXX XXX
07D6' 1A                        LD      A,(DE)
07D7' CD ** **          CALL    ISDEC
07DA' DA ** **          JP      C,ERR5
07DD' FE 38                     CP      '7'+1           ;0-7?
07DF' D2 ** **          JP      NC,ERR5
07E2' D6 30                     SUB     '0'
07E4' 87                        ADD     A,A
07E5' 87                        ADD     A,A
07E6' 87                        ADD     A,A
07E7' B1                        OR      C
07E8' 4F                        LD      C,A
07E9' 13                        INC     DE
07EA' CD ** **          CALL    CMACHK
07ED' 18 14                     JR      RL0
07EF'                   ;
07EF' 0E 28             _SRA:   LD      C,28H           ;00 101 XXX
07F1' 21                        DEFB    21H
07F2' 0E 20             _SLA:   LD      C,20H           ;00 100 XXX
07F4' 21                        DEFB    21H
07F5' 0E 38             _SRL:   LD      C,38H           ;00 111 XXX
07F7' 21                        DEFB    21H
07F8' 0E 08             _RRC:   LD      C,08H           ;00 001 XXX
07FA' 21                        DEFB    21H
07FB' 0E 00             _RLC:   LD      C,00H           ;00 000 XXX
07FD' 21                        DEFB    21H
07FE' 0E 18             _RR:    LD      C,18H           ;00 011 XXX
0800' 21                        DEFB    21H
0801' 0E 10             _RL:    LD      C,10H           ;00 010 XXX
0803' CD ** ** RL0:     CALL    REGCHK
0806' FE 0F                     CP      15              ;(IX+d)?
0808' 30 13                     JR      NC,RL[X]
080A' D6 05                     SUB     5
080C' DA ** **          JP      C,ERR5        ;rp
080F' FE 08                     CP      8
0811' D2 ** **          JP      NC,ERR5         ;index reg
0814' 47                        LD      B,A
0815' CD ** ** RLR:     CALL    PUTCB
0818' 78                        LD      A,B
0819' B1                        OR      C
081A' C3 ** **          JP      PUToA
081D'                   ;
081D' C5                RL[X]:  PUSH    BC
081E' CD ** **          CALL    PUT[X]
0821' CD ** **          CALL    PUTCB
0824' CD ** **          CALL    GETIXD
0827' CD ** **          CALL    PUToC
082A' C1                        POP     BC
082B' 79                        LD      A,C
082C' F6 06                     OR      06H             ;00 XXX 110
082E' C3 ** **          JP      PUToA
0831'                   ;
0831' AF                _RST:   XOR     A
0832' 32 ** **          LD      (UNDEF),A
0835' CD ** **          CALL    SPCUT
0838' CD ** **          CALL    DECI
083B' 24                        INC     H
083C' 25                        DEC     H
083D' C2 ** **          JP      NZ,ERR5
0840' 7D                        LD      A,L
0841' FE 08                     CP      08H
0843' 38 0E                     JR      C,RST0
0845' 6F                        LD      L,A
0846' E6 F8                     AND     0F8H            ;XXXXX000?
0848' BD                        CP      L               ;
0849' C2 ** **          JP      NZ,ERR5
084C' FE 40                     CP      40H             ;00-38H?
084E' 38 06                     JR      C,RST1
0850' C3 ** **          JP      ERR5
0853' 87                RST0:   ADD     A,A
0854' 87                        ADD     A,A
0855' 87                        ADD     A,A
0856' F6 C7             RST1:   OR      0C7H            ;11 000 111
0858' C3 ** **          JP      PUToA
085B'                   ;
085B' 1A                _IM:    LD      A,(DE)
085C' 13                        INC     DE
085D' D6 30                     SUB     '0'
085F' 28 0C                     JR      Z,IM0
0861' DA ** **          JP      C,ERR5
0864' FE 03                     CP      03H             ;1-2?
0866' D2 ** **          JP      NC,ERR5
0869' 3C                        INC     A
086A' 87                        ADD     A,A
086B' 87                        ADD     A,A
086C' 87                        ADD     A,A
086D' F6 46             IM0:    OR      046H            ;01 000 110
086F' 47                        LD      B,A
```

▶ X68000でリングマスターをやったことがあるのならわかると思うが，洞窟内でのBGMは耳について眠れない。へぶげぶ。
神田　望（18）東京都

```
0870' 0E ED                     LD      C,0EDH
0872' C3 ** **          JP      PUToBC
0875'                   ;
0875' 3A ** ** _LIST:   LD      A,(prname)
0878' 32 ** **          LD      (LSTSW),A     ;LSTSW on ( if prname is not present
ed)
087B' C9                        RET
087C'                   ;
087C' AF                _NLIST: XOR     A
087D' 32 ** **          LD      (LSTSW),A     ;LSTSW off
0880' C9                        RET
0881'                   ;
0881' CD ** ** _PAGE:   CALL    EVAL
0884' 3A ** **          LD      A,(UNDEF)
0887' E6 01                     AND     1
0889' C2 ** **          JP      NZ,ERR15      ;Value Error
088C'
088C' ED 43 ** **               LD      (PAGE),BC
0890' C9                        RET
0891'                   ;
0891' CD ** ** _TITLE:  CALL    SPCUT
0894' 06 50                     LD      B,80
0896' 21 ** **          LD      HL,TITLBF
0899' 1A                        LD      A,(DE)
089A' 13                        INC     DE
089B' FE 27                     CP      "'"
089D' 28 10                     JR      Z,TTL2
089F' FE 22                     CP      '"'
08A1' 28 1C                     JR      Z,TTL3
08A3'
08A3' 77                TTL1:   LD      (HL),A
08A4' 23                        INC     HL
08A5' 1A                        LD      A,(DE)
08A6' 13                        INC     DE
08A7' FE 0D                     CP      CR
08A9' 28 22                     JR      Z,TTL4
08AB' 10 F6                     DJNZ    TTL1
08AD' 18 1E                     JR      TTL4
08AF'
08AF' 1A                TTL2:   LD      A,(DE)
08B0' 13                        INC     DE
08B1' FE 27                     CP      "'"
08B3' 28 18                     JR      Z,TTL4
08B5' FE 0D                     CP      CR
08B7' 28 14                     JR      Z,TTL4
08B9' 77                        LD      (HL),A
08BA' 23                        INC     HL
08BB' 10 F2                     DJNZ    TTL2
08BD' 18 0E                     JR      TTL4
08BF'
08BF' 1A                TTL3:   LD      A,(DE)
08C0' 13                        INC     DE
08C1' FE 22                     CP      '"'
08C3' 28 08                     JR      Z,TTL4
08C5' FE 0D                     CP      CR
08C7' 28 04                     JR      Z,TTL4
08C9' 77                        LD      (HL),A
08CA' 23                        INC     HL
08CB' 10 F2                     DJNZ    TTL3
08CD'
08CD' 36 00             TTL4:   LD      (HL),0
08CF' C9                        RET
08D0'                   ;
08D0' 3E 01             _KANJI: LD      A,1
08D2' 32 ** **          LD      (KNJSW),A
08D5' C9                        RET
08D6'                   ;
08D6' AF                _NKANJI:XOR     A
08D7' 32 ** **          LD      (KNJSW),A
08DA' C9                        RET
08DB'                   ;
08DB' 3A ** ** _ORG:    LD      A,(prname)
08DE' A7                        AND     A
08DF' C4 ** **          CALL    NZ,PUTHD3
08E2'
08E2' CD ** **          CALL    EVAL
08E5' CD ** **          CALL    PUTeF
08E8' 3A ** **          LD      A,(UNDEF)
08EB' E6 01                     AND     1
08ED' C2 ** **          JP      NZ,ERR15      ;Value Error
08F0' ED 43 ** **               LD      (PTRFC),BC
08F4' 3A ** **          LD      A,(LOCFLG)
08F7' F6 01                     OR      1
08F9' 32 ** **          LD      (LOCFLG),A
08FC' 6F                        LD      L,A
08FD'
08FD' 3E E8                     LD      A,0E8H
08FF' CD ** **          CALL    PUTrA
0902'
0902' 3A ** **          LD      A,(SEGMD)
0905' FE 02                     CP      2
0907' 20 09                     JR      NZ,ORG1
0909'
0909' ED 43 ** **               LD      (PTRCD),BC    ;CSEG
090D' 7D                        LD      A,L
090E' 32 ** **          LD      (LOCCD),A
0911' C9                        RET
0912'
0912' FE 03             ORG1:   CP      3
0914' 20 09                     JR      NZ,ORG2
0916'
0916' ED 43 ** **               LD      (PTRDT),BC    ;DSEG
091A' 7D                        LD      A,L
091B' 32 ** **          LD      (LOCDT),A
091E' C9                        RET
091F'
091F' ED 43 ** **       ORG2:   LD      (PTRWK),BC
0923' 7D                        LD      A,L
0924' 32 ** **          LD      (LOCDT),A
0927' C9                        RET
0928'
0928'
0928' 3A ** ** _PHASE:  LD      A,(prname)
092B' A7                        AND     A
092C' C4 ** **          CALL    NZ,PUTHD3
092F'
092F' CD ** **          CALL    EVAL
0932' CD ** **          CALL    PUTeF
0935'
0935' 3A ** **          LD      A,(UNDEF)
0938' E6 01                     AND     1
093A' C2 ** **          JP      NZ,ERR15      ;Value Error
093D' ED 43 ** **               LD      (PTRFC),BC
0941'
0941' 3A ** **          LD      A,(LOCFLG)
0944' F6 02                     OR      2
0946' 32 ** **          LD      (LOCFLG),A
0949' 3E E2                     LD      A,0E2H
094B' CD ** **          CALL    PUTrA
094E' C9                        RET
094F'                   ;
094F'                   _DEPHASE:
094F' 3A ** **          LD      A,(prname)
0952' A7                        AND     A
0953' C4 ** **          CALL    NZ,PUTHD3
0956'
0956' 3A ** **          LD      A,(LOCFLG)
0959' E6 FD                     AND     0FDH
095B' 32 ** **          LD      (LOCFLG),A
095E' 3E E3                     LD      A,0E3H
0960' CD ** **          CALL    PUTrA
0963' C9                        RET
0964'                   ;
0964' 3A ** ** _EXT:    LD      A,(pass)
0967' A7                        AND     A
0968' 20 15                     JR      NZ,EXT2
096A'
096A' CD ** ** EXT1:    CALL    SPCUT         ;For PASS-1
096D' 3E 05                     LD      A,5           ;EXT
096F' 2A ** **          LD      HL,(LBLNUM)
0972' 23                        INC     HL
0973' 22 ** **          LD      (LBLNUM),HL
0976' CD ** **          CALL    DEFLBL
0979' CD ** **          CALL    ISSPRT
097C' 28 EC                     JR      Z,EXT1
097E' C9                        RET
097F'
097F' 3A ** ** EXT2:    LD      A,(prname)    ;For PASS-2
0982' A7                        AND     A
0983' C4 ** **          CALL    NZ,PUTHD3
0986'
0986' CD ** **          CALL    SEALBL
0989' A7                        AND     A
098A' C4 ** **          CALL    NZ,EXTLBL
098D' CD ** **          CALL    ISSPRT
0990' 28 ED                     JR      Z,EXT2
0992' C9                        RET
0993'
0993' ED 5B ** **       EXTLBL::LD      DE,(SV_PTR)
0997' CD ** **          CALL    LENLBL
099A' 78                        LD      A,B
099B' FE 20                     CP      31+1
099D' D2 ** **          JP      NC,ERR3               ;Too Long Label Err
09A0' F6 20                     OR      20H
09A2' CD ** **          CALL    PUTrA
09A5'
09A5' 1A                EXT3:   LD      A,(DE)
09A6' CD ** **          CALL    ISALNUM
09A9' 38 06                     JR      C,EXT4
09AB' 13                        INC     DE
09AC' CD ** **          CALL    PUTrA
09AF' 18 F4                     JR      EXT3
09B1'
09B1' 3E 81             EXT4:   LD      A,81H
09B3' CD ** **          CALL    PUTrAHL
09B6' 2A ** **          LD      HL,(SV_STR)
09B9' 23                        INC     HL
09BA' 23                        INC     HL
09BB' AF                        XOR     A
09BC' CD 9A 1F          CALL    _POKE
09BF' C9                        RET
09C0'                   ;
09C0' 3A ** ** _PUBLIC:LD       A,(pass)
09C3' A7                        AND     A
09C4' CA ** **          JP      Z,NXLIN
09C7'
09C7' CD ** ** PUB1:    CALL    SPCUT         ;For PASS-2
09CA' D5                        PUSH    DE
09CB' CD ** **          CALL    SEALBL
09CE' DA ** **          JP      C,ERR1        ;Undefined Label Flag
09D1' D1                        POP     DE
09D2'
09D2' FE 01                     CP      1
09D4' 20 13                     JR      NZ,PUB4
09D6' 60                        LD      H,B
09D7' 69                        LD      L,C
09D8' 3E E7                     LD      A,0E7H
09DA' CD ** **          CALL    PUTrAHL
09DD' 2A ** **          LD      HL,(LBLNUM)
09E0' 23                        INC     HL
09E1' 22 ** **          LD      (LBLNUM),HL
09E4' 3E 81                     LD      A,81H
09E6' CD ** **          CALL    PUTrAHL
09E9'
09E9' CD ** ** PUB4:    CALL    LENLBL
09EC' 78                        LD      A,B
09ED' FE 20                     CP      31+1
09EF' D2 ** **          JP      NC,ERR3               ;Too Long Label Name
09F2'
09F2' CD ** **          CALL    PUTrAHL
09F5' 1A                PUB2:   LD      A,(DE)
09F6' CD ** **          CALL    ISALNUM
09F9' 38 06                     JR      C,PUB3
09FB' 13                        INC     DE
09FC' CD ** **          CALL    PUTrA
09FF' 18 F4                     JR      PUB2
0A01' CD ** ** PUB3:    CALL    ISSPRT
0A04' 28 C1                     JR      Z,PUB1
0A06' C9                        RET
0A07'                   ;
0A07' 3E 01             _END:   LD      A,1
0A09' 32 ** **          LD      (endflg),A
0A0C'
0A0C' CD ** **          CALL    EOLCHK
0A0F' C8                        RET     Z
0A10' CD ** **          CALL    EVAL
0A13' CD ** **          CALL    PUTeF
0A16' 3E E9                     LD      A,0E9H
0A18' CD ** **          CALL    PUTrA
0A1B' C9                        RET
0A1C'
0A1C'
0A1C'                   ;
0A1C' 3A ** ** _CSEG:   LD      A,(LOCFLG)
0A1F' E6 02                     AND     2
0A21' C2 ** **          JP      NZ,ERR16      ;Not Enable bitween PHASE
0A24' 3E 02                     LD      A,2
0A26' 32 ** **          LD      (SEGMD),A
0A29' 3A ** **          LD      A,(LOCCD)
0A2C' 32 ** **          LD      (LOCFLG),A
0A2F' 2A ** **          LD      HL,(PTRCD)
0A32' 22 ** **          LD      (PTRFC),HL
0A35' 3E E1                     LD      A,0E1H
0A37' CD ** **          CALL    PUTrA
0A3A' 3E 00                     LD      A,0
0A3C' CD ** **          CALL    PUTrA
0A3F' C9                        RET
0A40'                   ;
0A40' 3A ** ** _DSEG:   LD      A,(LOCFLG)
0A43' E6 02                     AND     2
0A45' C2 ** **          JP      NZ,ERR16      ;Not Enable between PHASE
0A48' 3E 03                     LD      A,3
0A4A' 32 ** **          LD      (SEGMD),A
0A4D' 3A ** **          LD      A,(LOCDT)
0A50' 32 ** **          LD      (LOCFLG),A
0A53' 2A ** **          LD      HL,(PTRDT)
0A56' 22 ** **          LD      (PTRFC),HL
0A59' 3E E1                     LD      A,0E1H
0A5B' CD ** **          CALL    PUTrA
0A5E' 3E 01                     LD      A,1
0A60' CD ** **          CALL    PUTrA
0A63' C9                        RET
0A64'                   ;
0A64' 3A ** ** _COMMON:LD       A,(LOCFLG)
0A67' E6 02                     AND     2
0A69' C2 ** **          JP      NZ,ERR16      ;Not Enable between PHASE
0A6C' 3E 04                     LD      A,4
0A6E' 32 ** **          LD      (SEGMD),A
0A71' 3A ** **          LD      A,(LOCWK)
0A74' 32 ** **          LD      (LOCFLG),A
0A77' 2A ** **          LD      HL,(PTRWK)
0A7A' 22 ** **          LD      (PTRFC),HL
0A7D' 3E E1                     LD      A,0E1H
0A7F' CD ** **          CALL    PUTrA
0A82' 3E 02                     LD      A,2
0A84' CD ** **          CALL    PUTrA
0A87' C9                        RET
0A88'                   ;
0A88' CD ** ** _DEFB:   CALL    SPCUT
0A8B' 1A                        LD      A,(DE)
0A8C' CD ** **          CALL    ISQ
0A8F' 20 05                     JR      NZ,DEFB0
0A91' CD ** **          CALL    DM0
0A94' 18 06                     JR      DEFB2
0A96'
0A96' CD ** ** DEFB0:   CALL    EVAL
0A99' CD ** **          CALL    PUTe84
0A9C' CD ** ** DEFB2:   CALL    ISSPRT
0A9F' 28 E7                     JR      Z,_DEFB
0AA1' C9                        RET
0AA2'
0AA2'                   ;
0AA2' CD ** ** _DEFW:   CALL    SPCUT
0AA5' CD ** **          CALL    EVAL
0AA8' 3A ** **          LD      A,(UNDEF)
0AAB' E6 01                     AND     1
0AAD' 20 0A                     JR      NZ,_DEFW1
0AAF'
0AAF' 79                        LD      A,C
0AB0' CD ** **          CALL    PUToA
0AB3' 78                        LD      A,B
0AB4' CD ** **          CALL    PUToA
0AB7' 18 08                     JR      _DEFW2
0AB9'
0AB9' CD ** ** _DEFW1:  CALL    PUTeF
0ABC' 3E E5                     LD      A,0E5H
0ABE' CD ** **          CALL    PUTrA
0AC1' CD ** ** _DEFW2:  CALL    ISSPRT
0AC4' 28 DC                     JR      Z,_DEFW
0AC6' C9                        RET
0AC7'                   ;
0AC7' 1A                _DM:    LD      A,(DE)
0AC8' CD ** **          CALL    ISQ
0ACB' C2 ** **          JP      NZ,ERR12
0ACE' 3A ** ** DM0:     LD      A,(KNJSW)
0AD1' A7                        AND     A
0AD2' 20 28                     JR      NZ,DMK0       ;K+
0AD4'                           ;               put string if K-
0AD4' 13                DM1:    INC     DE
0AD5' 1A                        LD      A,(DE)
0AD6' FE 0D                     CP      CR
0AD8' CA ** **          JP      Z,ERR12
0ADB' FE 5E                     CP      '^'
0ADD' 28 18                     JR      Z,DM4
0ADF' CD ** **          CALL    ISQ
0AE2' 20 0E                     JR      NZ,DM3
0AE4' 13                        INC     DE
0AE5' 4F                        LD      C,A
0AE6' 1A                        LD      A,(DE)
0AE7' CD ** **          CALL    ISQ
0AEA' 28 04                     JR      Z,DM2
0AEC' CD ** **          CALL    ISLSPR
0AEF' C8                        RET     Z
0AF0' 1B                DM2:    DEC     DE
0AF1' 79                        LD      A,C
0AF2' CD ** ** DM3:     CALL    PUToA
0AF5' 18 DD                     JR      DM1
0AF7' CD ** ** DM4:     CALL    ESC
0AFA' 18 F6                     JR      DM3
0AFC'                   ;               put string if K+
0AFC' 13                DMK0:   INC     DE
0AFD' 1A                        LD      A,(DE)
0AFE' FE 0D                     CP      CR
0B00' CA ** **          JP      Z,ERR12
0B03' FE 5E                     CP      '^'
0B05' 28 2C                     JR      Z,DMK5
0B07' CD ** **          CALL    ISKNJ
0B0A' 38 0F                     JR      C,DMK2
0B0C' CD ** **          CALL    PUToA
0B0F' 13                        INC     DE
0B10' 1A                        LD      A,(DE)
0B11' FE 0D                     CP      CR
0B13' CA ** **          JP      Z,ERR12
0B16' CD ** **          CALL    PUToA
0B19' 18 E1                     JR      DMK0
0B1B' CD ** ** DMK2:    CALL    ISQ
0B1E' 20 0E                     JR      NZ,DMK4
0B20' 13                        INC     DE
0B21' 4F                        LD      C,A
0B22' 1A                        LD      A,(DE)
0B23' CD ** **          CALL    ISQ
0B26' 28 04                     JR      Z,DMK3
0B28' CD ** **          CALL    ISLSPR
0B2B' C8                        RET     Z
0B2C' 1B                DMK3:   DEC     DE
0B2D' 79                        LD      A,C
0B2E' CD ** ** DMK4:    CALL    PUToA
0B31' 18 C9                     JR      DMK0
0B33' CD ** ** DMK5:    CALL    ESC
0B36' 18 F6                     JR      DMK4
0B38'                   ;
0B38' 13                ESC::   INC     DE
0B39' 1A                        LD      A,(DE)
0B3A' FE 0D                     CP      CR
0B3C' CA ** **          JP      Z,ERR12
0B3F' D6 40                     SUB     '@'
0B41' 38 03                     JR      C,ESC0
0B43' FE 1C                     CP      '['-'@'+1
0B45' D8                        RET     C
0B46' C6 40             ESC0:   ADD     A,'@'
0B48' C9                        RET
0B49'                   ;
0B49' CD ** ** _DEFS:   CALL    EVAL
0B4C' CD ** **          CALL    PUTeF
0B4F' 3E B0                     LD      A,0B0H
0B51' CD ** **          CALL    PUTrA
0B54'
0B54' 3A ** **          LD      A,(UNDEF)
0B57' E6 01                     AND     1
0B59' C2 ** **          JP      NZ,ERR15      ;Value Err
0B5C'
0B5C' 60                        LD      H,B
0B5D' 69                        LD      L,C
0B5E' CD ** **          CALL    ADDPC
0B61' C9                        RET
0B62'                   ;
0B62'                   ;       check whether kanji 1st char
0B62'                   ;
0B62' FE 81             ISKNJ:  CP      81H
0B64' D8                        RET     C
0B65' FE A0                     CP      0A0H
0B67' 3F                        CCF
0B68' D0                        RET     NC
0B69' FE E0                     CP      0E0H
0B6B' D8                        RET     C
0B6C' FE F0                     CP      0F0H
0B6E' 3F                        CCF
0B6F' C9                        RET
0B70'                   ;
0B70'                   ;       skip white space
0B70'                   ;
0B70' 13                SPCUT0: INC     DE
0B71' 1A                SPCUT:: LD      A,(DE)
0B72' FE 09                     CP      TAB
0B74' 28 FA                     JR      Z,SPCUT0
0B76' FE 20                     CP      SPC
0B78' 28 F6                     JR      Z,SPCUT0
0B7A' C9                        RET
0B7B'                   ;
0B7B'                   ;       check whether separator
0B7B'                   ;
0B7B' CD ** ** ISSPRT:  CALL    SPCUT
0B7E' FE 2C                     CP      ','
0B80' 28 03                     JR      Z,ISSPRT1
0B82' FE 3A                     CP      ':'
0B84' C0                        RET     NZ
0B85'
0B85' 13                ISSPRT1:INC     DE
0B86' C9                        RET
0B87'                   ;
0B87'                   ISLSPR0::
0B87' 1A                        LD      A,(DE)
0B88' D9                ISLSPR: EXX
0B89' 21 ** **          LD      HL,LSPTBL
0B8C' 01 ** **          LD      BC,SPREND-LSPTBL
0B8F' ED B1                     CPIR
0B91' D9                        EXX
0B92' C9                        RET
0B93'                   ;
0B93'                   ;       check whether blank
0B93'                   ;
0B93' 1A                ISBLK0: LD      A,(DE)
0B94' FE 09             ISBLNK: CP      TAB
0B96' C8                        RET     Z
0B97' FE 0D                     CP      CR
0B99' C8                        RET     Z
0B9A' FE 3B                     CP      ';'
0B9C' C8                        RET     Z
0B9D' FE 20                     CP      SPC
0B9F' C9                        RET
0BA0'                   ;
0BA0'                   ;       check whether quotation mark
0BA0'                   ;
0BA0' FE 27             ISQ::   CP      "'"
0BA2' C8                        RET     Z
0BA3' FE 22                     CP      '"'
0BA5' C9                        RET
0BA6'                   ;
0BA6'                   ;       get condition code no
0BA6'                   ;
0BA6' 21 ** ** GETCND:  LD      HL,CNDTBL
0BA9' 06 08                     LD      B,8
0BAB' 18 13                     JR      GTREG0
0BAD'                   ;
0BAD' CD ** ** REGCHK:  CALL    GETREG
0BB0' D0                        RET     NC
0BB1' C3 ** **          JP      ERR5
0BB4'                   ;
0BB4'                   ;       get register no
0BB4'                   ;
0BB4' 21 ** ** GETREG:  LD      HL,REGTBL
0BB7' 06 11                     LD      B,17
0BB9' 18 05                     JR      GTREG0
0BBB'                   ;
0BBB'                   ;       get 8bit register no
0BBB'                   ;
0BBB' 21 ** ** GETRG8:  LD      HL,RG8TBL
0BBE' 06 08                     LD      B,8
0BC0' C5                GTREG0: PUSH    BC
0BC1' 48                        LD      C,B
0BC2' D5                GTREG1: PUSH    DE
0BC3' 1A                GTREG2: LD      A,(DE)
0BC4' BE                        CP      (HL)
0BC5' 20 04                     JR      NZ,GTREG3
0BC7' 13                        INC     DE
0BC8' 23                        INC     HL
0BC9' 18 F8                     JR      GTREG2
0BCB' 34                GTREG3: INC     (HL)
0BCC' 35                        DEC     (HL)
0BCD' 20 0A                     JR      NZ,GTREG4
0BCF' CD ** **          CALL    ISLSPR
0BD2' 20 05                     JR      NZ,GTREG4
0BD4' E1                        POP     HL
0BD5' 79                        LD      A,C
0BD6' 90                        SUB     B
0BD7' C1                        POP     BC
0BD8' C9                        RET                   ;NC
0BD9' CD ** ** GTREG4:  CALL    SEAZ
0BDC' D1                        POP     DE
0BDD' 10 E3                     DJNZ    GTREG1
0BDF' C1                        POP     BC
0BE0' 37                        SCF
0BE1' C9                        RET
0BE2'                   ;
0BE2' 34                CMPGEN: INC     (HL)
0BE3' 35                        DEC     (HL)
```

▶今，私は，中間テストの真っただ中にいる。この私が勉強するわけは，薬学部に受かりたいからだ！　なぜ薬学部かっていうと，薬剤師になって病院に入って院長の娘と結ばれ逆玉の輿を目指しているからさ！　フン。これがおれの，7ヵ年計画だー！　これからを見守っていてくれ！

塩川　健一（17）静岡県

```
0BE4' 37                        SCF
0BE5' C8                        RET    Z
0BE6' 4B                        LD     C,E
0BE7' 42                        LD     B,D
0BE8' 1A               CPGEN0:  LD     A,(DE)
0BE9' BE                        CP     (HL)
0BEA' 20 05                     JR     NZ,CPGEN1
0BEC' 13                        INC    DE
0BED' 23                        INC    HL
0BEE' C3 ** **         JP       CPGEN0
0BF1' 34               CPGEN1:  INC    (HL)
0BF2' 35                        DEC    (HL)
0BF3' 23                        INC    HL
0BF4' 20 05                     JR     NZ,CPGEN2
0BF6' CD ** **         CALL     ISBLNK
0BF9' 28 0A                     JR     Z,CPGEN3
0BFB' CD ** ** CPGEN2: CALL     SEAZ
0BFE' 23                        INC    HL
0BFF' 23                        INC    HL
0C00' 59                        LD     E,C
0C01' 50                        LD     D,B
0C02' C3 ** **         JP       CMPGEN
0C05' 4E               CPGEN3:  LD     C,(HL)
0C06' 23                        INC    HL
0C07' 46                        LD     B,(HL)
0C08' 78                        LD     A,B
0C09' B7                        OR     A
0C0A' 20 05                     JR     NZ,CPGEN4
0C0C' CD ** **         CALL     PUToC
0C0F' B7                        OR     A
0C10' C9                        RET
0C11' CD ** ** CPGEN4: CALL     PUToBC
0C14' B7                        OR     A
0C15' C9                        RET
0C16'                  ;
0C16'                  ;
0C16'                  ;        skip to next eos
0C16'                  ;
0C16' AF               SEAZ:    XOR    A
0C17' BE               SEAZ0:   CP     (HL)
0C18' 23                        INC    HL
0C19' C8                        RET    Z
0C1A' C3 ** **         JP       SEAZ0
0C1D'                  ;
0C1D'                  ;        get displacement
0C1D'                  ;             of index addressing
0C1D'                  ;
0C1D' CD ** ** GETXD0: CALL     EVAL]
0C20' AF                        XOR    A
0C21' 91                        SUB    C
0C22' 4F                        LD     C,A
0C23' C9                        RET
```

```
0C24'
0C24' CD ** ** PUT[X]D:CALL     PUT[X]
0C27' 1A               GETIXD:  LD     A,(DE)
0C28' 13                        INC    DE
0C29' FE 2B                     CP     '+'
0C2B' 28 10                     JR     Z,EVAL]
0C2D' FE 2D                     CP     '-'
0C2F' 28 EC                     JR     Z,GETXD0
0C31' FE 29                     CP     ')'
0C33' C2 ** **         JP       NZ,ERR5
0C36' AF                        XOR    A
0C37' 32 ** **         LD       (UNDEF),A
0C3A' 0E 00                     LD     C,0
0C3C' C9                        RET
0C3D'                           .
0C3D'                  ;
0C3D'                  ;        evaluate 'expression)'
0C3D'                  ;
0C3D' CD ** ** EVAL]:  CALL     EVAL
0C40'                           ;
0C40' 1A                        LD     A,(DE)
0C41' FE 29                     CP     ')'
0C43' C2 ** **         JP       NZ,ERR10
0C46' 13                        INC    DE
0C47' C9                        RET
0C48'                  ;
0C48'                  ;        evaluate '(expression)'
0C48'                  ;
0C48' CD ** ** [EVAL]: CALL     IS[
0C4B' C2 ** **         JP       NZ,ERR5
0C4E' 18 ED                     JR     EVAL]
0C50'                  ;
0C50'                  ;        check ','
0C50'                  ;
0C50' 47               CMACHK:  LD     B,A
0C51' 1A                        LD     A,(DE)
0C52' FE 2C                     CP     ','
0C54' 78                        LD     A,B
0C55' C2 ** **         JP       NZ,ERR11
0C58' 13                        INC    DE
0C59' C9                        RET
0C5A'                  ;
0C5A'                  ;        check '('
0C5A'                  ;
0C5A' 1A               IS[:     LD     A,(DE)
0C5B' FE 28                     CP     '('
0C5D' C0                        RET    NZ
0C5E' 13                        INC    DE
0C5F' C9                        RET
0C60'                  ;
0C60'                  ;        check whether line pointer is
0C60'                  ;                 at end of line
```

```
0C60'                  ;
0C60' CD ** ** EOLCHK: CALL     SPCUT
0C63' FE 0D                     CP     CR
0C65' C8                        RET    Z
0C66' FE 3B                     CP     ';'
0C68' C9                        RET
0C69'
0C69'                  ;
0C69'                  ;        works
0C69'                  ;
0C69'                           DSEG
0000"
0000"                  pass::   DEFS   1
0001"                  SPBUF::  DEFS   2      ;sp reserve buffer
0003"                  LINPTR::DEFS    2      ;current line pointer
0005"                  LINNO::  DEFS   2      ;current line no
0007"                  LBLPTR::DEFS    2      ;current label pointer
0009"                  WKMAX:   DEFS   2      ;end oflabel table
000B"                  PADR::   DEFS   2      ;address for print out
000D"                  LBLNUM::DEFS    2      ;; LABEL NUMBER
000F"                  ERRCNT::DEFS    2      ;Counter of Error
0011"                  LSTSW::  DEFS   1      ;listing on/offswitch
0012"                  KNJSW::  DEFS   1      ;whether KANJI is used
0013"
0013"                  LOCFLG::DS      1      ;Attribute of Location
0014"                  LOCCD::  DS     1      ;LOCFLGfor CSEG
0015"                  LOCDT::  DS     1      ;LOCFLGfor DSEG
0016"                  LOCWK::  DS     1      ;LOCFLGfor WSEG
0017"
0017"                  SEGMD::  DS     1      ;Segment mode
0018"                  PTRCD::  DS     2      ;PC forcode segment
001A"                  PTRDT::  DS     2      ;PC fordata segment
001C"                  PTRWK::  DS     2      ;PC forwork segment
001E"                  PTRFC::  DS     2      ;Current PC
0020"
0020"                  JRSEG::  DS     1      ;The segment ofthe expression for JR
&     DJNZ
0021"
0021"                  endflg::DEFS    1      ;If endflg = 1,All ends
0022"
0022"                  CNTbA::  DEFS   1
0023"                  CNTeA::  DEFS   1
0024"                  LSTOBJ::DEFS    1
0025"                  BFOBJ::  DEFS   32     ;Buffers for object code
0045"                  BFEVAL::DEFS    50     ;Buffers for routine-EVAL
0077"                  UNDEF::  DEFS   1      ;Whether LABEL is used ?
0078"
0078"                  PAGE::   DEFS   2      ;The number of lines per page
007A"                  CNTLINE::DEFS   2
007C"                  TITLBF::DEFS    81     ;TITLE String Buffer
00CD"
00CD"                           END
```

リスト5 WZDソースリスト4

```
0000'                  ;==========================================
0000'                  ; WZD Z80 Relocatable Asembler
0000'                  ;         Module Part 3.5
0000'                  ;         The Routines    DevidedFrom WZD3.ASM
0000'                  ;         Programed By    T.Ishigami
0000'                  ;              '90 May 26th
0000'                  ;
0000'                  ;==========================================
0000'
0000'                         .LIST
0000'                  ;
0000'                  ; Externals
0000'                  ;
                              EXT    UNDEF
                              EXT    JRSEG
                              EXT    CNTeA
                              EXT    PUTeA,PUTeAHL
                              EXT    pass
                              EXT    SPBUF
                              EXT    SEALBL
                              EXT    LINPTR
                              EXT    NXLIN
                              EXT    LINNO
                              EXT    ISQ
                              EXT    ERRCNT
                              EXT    SPCUT
                              EXT    PADR
                              EXT    SEGMD
                              EXT    GETADRS
                              EXT    ISDEC
                              EXT    LENLBL
                              EXT    SKPLBL
                              EXT    EXTLBL
                              EXT    ISLSPR0
                              EXT    ESC
                              EXT    ISALNUM
                              EXT    PTRCD
                              EXT    PTRDT
                              EXT    PTRWK
                              EXT    PTRFC
0000'
0000'                  ;
0000'                  ; Constans
0000'                  ;
0009                   TAB      EQU    9
000D                   CR       EQU    0DH
0020                   SPC      EQU    20H
0000'                  ;
0000'                  ;== Initialize Work Areas ==
0000'                  ;
0000' 21 00 00 iniwk1::LD       HL,0
0003' 22 ** **         LD       (LBLNUM##),HL
0006' 22 ** **         LD       (CNTLINE##),HL
0009' 22 ** **         LD       (ERRCNT##),HL
000C' 21 00 02         LD       HL,200H
000F' 22 ** **         LD       (LBLPTR##),HL
0012' 21 FF FF         LD       HL,-1
0015' 22 ** **         LD       (PAGE##),HL
0018' AF                        XOR    A
0019' 32 ** **         LD       (TITLBF##),A
001C'                  ;
001C'                  ;== Initialize work areas Part II ==
001C'                  ;
001C' AF               iniwk2::XOR     A
001D' 32 ** **         LD       (CNTbA##),A
0020' 32 ** **         LD       (LSTOBJ##),A
0023' 32 ** **         LD       (LOCCD##),A
0026' 32 ** **         LD       (LOCDT##),A
0029' 32 ** **         LD       (LOCWK##),A
002C' 32 ** **         LD       (LOCFLG##),A
002F' 3E 02                     LD     A,2
0031' 32 ** **         LD       (SEGMD),A    ;CSEG
0034'
0034' 3E 00                     LD     A,0
0036' 32 ** **         LD       (KNJSW##),A  ;Not Enable KANJI used
0039'
0039' 21 00 00         LD       HL,0
003C' 22 ** **         LD       (LINNO),HL
003F' 22 ** **         LD       (PTRCD),HL
0042' 22 ** **         LD       (PTRDT),HL
0045' 22 ** **         LD       (PTRWK),HL
0048' 22 ** **         LD       (PTRFC),HL
004B' C9                        RET
004C'                  ;
004C'                  ; Increment respectable PC
004C'                  ;
004C' F5               INCPC::  PUSH   AF
004D' E5                        PUSH   HL
004E' 3A ** **         LD       A,(SEGMD)
0051' 21 ** **         LD       HL,PTRCD
0054' FE 02                     CP     2
0056' 28 0A                     JR     Z,INCpc1
0058' 21 ** **         LD       HL,PTRDT
005B' FE 03                     CP     3
005D' 28 03                     JR     Z,INCpc1
005F' 21 ** **         LD       HL,PTRWK
0062' 34               INCpc1:  INC    (HL)
0063' 20 02                     JR     NZ,INCpc2
0065' 23                        INC    HL
0066' 34                        INC    (HL)
0067' 2A ** ** INCpc2: LD       HL,(PTRFC)
006A' 23                        INC    HL
006B' 22 ** **         LD       (PTRFC),HL
006E' E1                        POP    HL
006F' F1                        POP    AF
0070' C9                        RET
0071'                  ;
0071'                  ; Add respectable PC with HL
```

```
0071'                  ;
0071' E5               ADDPC::  PUSH   HL
0072' ED 4B ** **               LD     BC,(PTRFC)
0076' 09                        ADD    HL,BC
0077' 22 ** **         LD       (PTRFC),HL
007A' E1                        POP    HL
007B'
007B' 3A ** **         LD       A,(SEGMD)
007E' FE 02                     CP     2
0080' 20 09                     JR     NZ,ADDPC1
0082'
0082' ED 4B ** **               LD     BC,(PTRCD)   ;CSEG
0086' 09                        ADD    HL,BC
0087' 22 ** **         LD       (PTRCD),HL
008A' C9                        RET
008B'
008B' FE 03            ADDPC1:  CP     3
008D' 20 09                     JR     NZ,ADDPC2
008F'
008F' ED 4B ** **               LD     BC,(PTRDT)   ;DSEG
0093' 09                        ADD    HL,BC
0094' 22 ** **         LD       (PTRDT),HL
0097' C9                        RET
0098'
0098' ED 4B ** **      ADDPC2:  LD     BC,(PTRWK)   ;WSEG
009C' 09                        ADD    HL,BC
009D' 22 ** **         LD       (PTRWK),HL
00A0' C9                        RET
00A1'
00A1'                  ;
00A1'                  ;        init hash table
00A1'                  ;
00A1' 21 00 00 inihsh::LD       HL,0
00A4' 01 01 02         LD       BC,200H      + 1
00A7' AF                        XOR    A
00A8' CD 9A 1F INIHS0: CALL     _POKE
00AB' ED A1                     CPI
00AD' EA ** **         JP       PE,INIHS0
00B0' C9                        RET
00B1'                  ;
00B1'                  ;        evaluate expression
00B1'                  ;
00B1' AF               EVAL::   XOR    A
00B2' 32 ** **         LD       (CNTeA),A
00B5' 32 ** **         LD       (UNDEF),A    ;Clear UNDEF
00B8' CD ** ** reEVAL: CALL     EVAL10
00BB' CD ** ** EVAL0:  CALL     SPCUT
00BE' FE 2B                     CP     '+'
00C0' 20 0A                     JR     NZ,EVAL1
00C2' E5                        PUSH   HL
00C3' CD ** **         CALL     EVAL9
00C6' C1                        POP    BC
00C7' CD ** **         CALL     _EVADD
00CA' 18 EF                     JR     EVAL0
00CC' FE 2D            EVAL1:   CP     '-'
00CE' 20 0C                     JR     NZ,EVAL2
00D0' E5                        PUSH   HL
00D1' CD ** **         CALL     EVAL9
00D4' 4D                        LD     C,L
00D5' 44                        LD     B,H
00D6' E1                        POP    HL
00D7' CD ** **         CALL     _EVSUB
00DA' 18 DF                     JR     EVAL0
00DC' 4D               EVAL2:   LD     C,L
00DD' 44                        LD     B,H
00DE' C9                        RET
00DF'                  ;
00DF' 13               EVAL9:   INC    DE
00E0' CD ** ** EVAL10: CALL     EVAL20
00E3' CD ** ** EVAL11: CALL     SPCUT
00E6' FE 2A                     CP     '*'
00E8' 20 0A                     JR     NZ,EVAL12
00EA' E5                        PUSH   HL
00EB' CD ** **         CALL     EVAL19
00EE' C1                        POP    BC
00EF' CD ** **         CALL     _EVMUL
00F2' 18 EF                     JR     EVAL11
00F4' FE 2F            EVAL12:  CP     '/'
00F6' 20 0C                     JR     NZ,EVAL13
00F8' E5                        PUSH   HL
00F9' CD ** **         CALL     EVAL19
00FC' 4D                        LD     C,L
00FD' 44                        LD     B,H
00FE' E1                        POP    HL
00FF' CD ** **         CALL     _EVDIV
0102' 18 DF                     JR     EVAL11
0104' FE 25            EVAL13:  CP     '%'
0106' C0                        RET    NZ
0107' E5                        PUSH   HL
0108' CD ** **         CALL     EVAL19
010B' 4D                        LD     C,L
010C' 44                        LD     B,H
010D' E1                        POP    HL
010E' D5                        PUSH   DE
010F' CD ** **         CALL     _EVMOD
0112' D1                        POP    DE
0113' 18 CE                     JR     EVAL11
0115'                  ;
0115' CD ** ** MINS:   CALL     EVAL19
0118' 7C                        LD     A,H
0119' 2F                        CPL
011A' 67                        LD     H,A
011B' 7D                        LD     A,L
011C' 2F                        CPL
011D' 6F                        LD     L,A
011E' 23                        INC    HL
011F' C9                        RET
0120'
```

```
0120' 4F               CHAR:    LD     C,A
0121' 13                        INC    DE
0122' 1A                        LD     A,(DE)
0123' FE 0D                     CP     CR
0125' CA ** **         JP       Z,ERR12
0128' FE 5E                     CP     '^'
012A' CC ** **         CALL     Z,ESC
012D' 26 00                     LD     H,0
012F' 6F                        LD     L,A
0130' 13                        INC    DE
0131' 1A                        LD     A,(DE)
0132' 13                        INC    DE
0133' B9                        CP     C
0134' C2 ** **         JP       NZ,ERR12
0137' C9                        RET
0138'                  ;
0138' 13               EVAL19:  INC    DE
0139' CD ** ** EVAL20: CALL     SPCUT
013C' FE 28                     CP     '('
013E' CA ** **         JP       Z,PAREN
0141' FE 2D                     CP     '-'
0143' CA ** **         JP       Z,_EVMINS
0146' FE 24                     CP     '$'
0148' CA ** **         JP       Z,_EVHEX
014B' FE 25                     CP     '%'
014D' CA ** **         JP       Z,_EVBIN
0150' CD ** **         CALL     ISDEC
0153' D2 ** **         JP       NC,_EVDECI
0156' CD ** **         CALL     ISQ
0159' CA ** **         JP       Z,_EVCHAR
015C' 21 ** **         LD       HL,ESPTBL
015F' 01 ** **         LD       BC,SPREND-ESPTBL
0162' ED B1                     CPIR
0164' CA ** **         JP       Z,ERR9
0167'                  ;
0167' C5               LABEL:   PUSH   BC
0168' 3A ** **         LD       A,(pass)
016B' A7                        AND    A
016C' 20 29                     JR     NZ,LABEL2
016E'
016E' 3A ** **         LD       A,(UNDEF)    ;For Pass-1
0171' F6 02                     OR     2           ;A Label is used
0173' 32 ** **         LD       (UNDEF),A
0176'
0176' CD ** **         CALL     ISEXT
0179' 28 10                     JR     Z,LABEL5
017B' CD ** **         CALL     SEALBL
017E' 38 04                     JR     C,LABEL1
0180' FE 01                     CP     1
0182' 28 39                     JR     Z,LABEL4
0184'
0184' 3E 03            LABEL1:  LD     A,3         ;Label is used
0186' 32 ** **         LD       (UNDEF),A   ;      &
0189' 18 32                     JR     LABEL4      ;Value is not certain
018B'
018B' 3E 03            LABEL5:  LD     A,3
018D' 32 ** **         LD       (UNDEF),A
0190' CD ** **         CALL     SKPLBL
0193' 13                        INC    DE
0194' 13                        INC    DE          ;SKIP '##'
0195' 18 26                     JR     LABEL4
0197'
0197'                  ;
0197' 3A ** ** LABEL2: LD       A,(UNDEF)    ;For PASS-2
019A' F6 02                     OR     2
019C' 32 ** **         LD       (UNDEF),A    ;A Label is used
019F'
019F' CD ** **         CALL     ISEXT
01A2' 28 26                     JR     Z,LABEL6
01A4' CD ** **         CALL     SEALBL
01A7' DA ** **         JP       C,ERR1       ;Undefined Label Error
01AA' FE 01                     CP     1           ;ASEG
01AC' 28 13                     JR     Z,LABEL3
01AE' FE 05                     CP     5           ;EXT
01B0' CA ** **         JP       Z,LABEL9
01B3'
01B3' 3E 03                     LD     A,3
01B5' 32 ** **         LD       (UNDEF),A
01B8' 3E 80                     LD     A,80H
01BA' CD ** **         CALL     PUTeAHL
01BD' 60               LABEL4:  LD     H,B
01BE' 69                        LD     L,C
01BF' C1                        POP    BC
01C0' C9                        RET
01C1'
01C1' 60               LABEL3:  LD     H,B
01C2' 69                        LD     L,C
01C3' 3E E7                     LD     A,0E7H
01C5' CD ** **         CALL     PUTeAHL
01C8' C1                        POP    BC
01C9' C9                        RET
01CA'
01CA' CD ** ** LABEL6: CALL     LENLBL
01CD' 78                        LD     A,B
01CE' FE 20                     CP     31+1
01D0' D2 ** **         JP       NC,ERR3            ;Too Long label Name
01D3' F6 20                     OR     20H
01D5' CD ** **         CALL     PUTeA
01D8' 1A               LABEL7:  LD     A,(DE)
01D9' CD ** **         CALL     ISALNUM
01DC' 38 06                     JR     C,LABEL8
01DE' 13                        INC    DE
01DF' CD ** **         CALL     PUTeA
01E2' 18 F4                     JR     LABEL7
01E4' 3E 03            LABEL8:  LD     A,3
01E6' 32 ** **         LD       (UNDEF),A
01E9' 13                        INC    DE
01EA' 13                        INC    DE          ;SKIP '##'
```

THE SENTINEL

▶先日のX68000ウイルス騒ぎで，X68000は一気にメジャーになってしまったようですね。しかし，TV画面で見るX68000の姿は一段と美しかったのは，いうまでもないことですね。

荒木　芳典（20）岡山県

```
01EB' C1                          POP   BC
01EC' C9                          RET
01ED'
01ED' 3E 03             LABEL9:   LD    A,3
01EF' 32 ** **          LD        (UNDEF),A
01F2' 3E 80                       LD    A,80H
01F4' CD ** **          CALL      PUTeAHL
01F7' CD ** **          CALL      EXTLBL
01FA' C1                          POP   BC
01FB' C9                          RET
01FC'
01FC'
01FC' C5                PAREN:    PUSH  BC
01FD' 13                          INC   DE
01FE' CD ** **          CALL      reEVAL
0201' 60                          LD    H,B
0202' 69                          LD    L,C
0203' C1                          POP   BC
0204' CD ** **          CALL      SPCUT
0207' 13                          INC   DE
0208' FE 29                       CP    ')'
020A' C2 ** **          JP        NZ,ERR10
020D' C9                          RET
020E'                   ;
020E'                   ; Whether The Label is external
020E'                   ;
020E' D5                ISEXT:    PUSH  DE
020F' CD ** **          CALL      SKPLBL
0212' 1A                          LD    A,(DE)
0213' 13                          INC   DE
0214' FE 23                       CP    '#'
0216' 20 04                       JR    NZ,ISDB1
0218' 1A                          LD    A,(DE)
0219' 13                          INC   DE
021A' FE 23                       CP    '#'
021C' D1                ISDB1:    POP   DE
021D' C9                          RET
021E'
021E'                   ;
021E' 13                JRHEX:    INC   DE
021F' 1A                          LD    A,(DE)
0220' CD B8 1F          CALL      _HEX
0223' D2 ** **          JP        NC,HEX0
0226'
0226' 2A ** **          LD        HL,(PADR)
0229'
0229' 3A ** **          LD        A,(UNDEF)
022C' F6 02                       OR    2
022E' 32 ** **          LD        (UNDEF),A
0231'
0231' 3A ** **          LD        A,(SEGMD)
0234' 47                          LD    B,A
0235' 3A ** **          LD        A,(JRSEG)
0238' A7                          AND   A
0239' 28 04                       JR    Z,JRHEX1
023B' B8                          CP    B
023C' C2 ** **          JP        NZ,ERR9             ;Illegal Expression
023F' 78                JRHEX1:   LD    A,B
0240' 32 ** **          LD        (JRSEG),A
0243' C9                          RET
0244'                   ;
0244' 13                HEX:      INC   DE
0245' 1A                          LD    A,(DE)
0246' CD B8 1F          CALL      _HEX
0249' 30 2B                       JR    NC,HEX0
024B'
024B' 2A ** **          LD        HL,(PADR)
024E' 3A ** **          LD        A,(SEGMD)
0251' FE 01                       CP    1             ;ASEG
0253' C8                          RET   Z
0254'
0254' D5                          PUSH  DE
0255' 3E 87                       LD    A,87H
0257' CD ** **          CALL      PUTeA
025A' CD ** **          CALL      GETADRS             ;ｹﾞﾝｻﾞｲﾉ PC ｦ ｱﾗﾜｽ '$' ﾉ ｼｮﾘ
025D' ED 5B ** **                 LD    DE,(PADR)
0261' B7                          OR    A
0262' ED 52                       SBC   HL,DE
0264' 3E E7                       LD    A,0E7H
0266' CD ** **          CALL      PUTeAHL
0269' 3E C1                       LD    A,0C1H        ;MINUS CODE
026B' CD ** **          CALL      PUTeA
026E' D1                          POP   DE
026F'
026F' 3E 03                       LD    A,3
0271' 32 ** **          LD        (UNDEF),A
0274' B7                          OR    A             ;CY = 0
0275' C9                          RET
0276'
0276' 6F                HEX0:     LD    L,A
0277' 26 00                       LD    H,0
0279' 13                          INC   DE
027A' 1A                HEX1:     LD    A,(DE)
027B' CD B8 1F          CALL      _HEX
027E' D8                          RET   C
027F' 29                          ADD   HL,HL
0280' 29                          ADD   HL,HL
0281' 29                          ADD   HL,HL
0282' 29                          ADD   HL,HL
0283' B5                          OR    L
0284' 6F                          LD    L,A
0285' 13                          INC   DE
0286' 18 F2                       JR    HEX1
0288'
0288'                   ;
0288' 21 00 00  BIN:    LD        HL,0
028B' 13                BIN0:     INC   DE
028C' 1A                          LD    A,(DE)
028D' FE 5F                       CP    '_'
028F' 28 FA                       JR    Z,BIN0
0291' D6 30             ISBIN:    SUB   '0'
0293' D8                          RET   C
0294' FE 02                       CP    2
0296' D0                          RET   NC
0297' 0F                          RRCA
0298' ED 6A                       ADC   HL,HL
029A' 18 EF                       JR    BIN0
029C'                   ;
029C' D5                DECI::    PUSH  DE
029D' D6 30                       SUB   '0'
029F' 4F                          LD    C,A
02A0' 13                DECI0:    INC   DE
02A1' CD ** **          CALL      ISLSPR0
02A4' 20 FA                       JR    NZ,DECI0
02A6' 1B                          DEC   DE
02A7' 1A                          LD    A,(DE)
02A8' D1                          POP   DE
02A9' FE 48                       CP    'H'
02AB' 20 0C                       JR    NZ,DECI1
02AD' 79                          LD    A,C
02AE' CD ** **          CALL      HEX0
02B1' 1A                          LD    A,(DE)
02B2' FE 48                       CP    'H'
02B4' C2 ** **          JP        NZ,ERR9
02B7' 13                          INC   DE
02B8' C9                          RET
02B9' FE 42             DECI1:    CP    'B'
02BB' 20 14                       JR    NZ,DECI2
02BD' 79                          LD    A,C
02BE' FE 02                       CP    2
02C0' D2 ** **          JP        NC,ERR9
02C3' 6F                          LD    L,A
02C4' 26 00                       LD    H,0
02C6' CD ** **          CALL      BIN0
02C9' 1A                          LD    A,(DE)
02CA' FE 42                       CP    'B'
02CC' C2 ** **          JP        NZ,ERR9
02CF' 13                          INC   DE
02D0' C9                          RET
02D1' 69                DECI2:    LD    L,C
02D2' 26 00                       LD    H,0
02D4' 13                DECI3:    INC   DE
02D5' 1A                          LD    A,(DE)
02D6' CD ** **          CALL      ISDEC
02D9' D8                          RET   C
02DA' D6 30                       SUB   '0'
02DC' 4D                          LD    C,L
02DD' 44                          LD    B,H
02DE' 29                          ADD   HL,HL
02DF' 29                          ADD   HL,HL
02E0' 09                          ADD   HL,BC
02E1' 29                          ADD   HL,HL
02E2' 4F                          LD    C,A
02E3' 06 00                       LD    B,0
02E5' 09                          ADD   HL,BC
02E6' 18 EC                       JR    DECI3
02E8'
02E8' 09                _EVADD:   ADD   HL,BC
02E9' 3E C0                       LD    A,0C0H ;ADD CODE
02EB' CD ** **          CALL      PUTeA
02EE' C9                          RET
02EF'
02EF' B7                _EVSUB:   OR    A
02F0' ED 42                       SBC   HL,BC
02F2' 3E C1                       LD    A,0C1H ;SUB CODE
02F4' CD ** **          CALL      PUTeA
02F7' C9                          RET
02F8'
02F8' 3A ** **  _EVMUL: LD        A,(UNDEF)
02FB' E6 02                       AND   2
02FD' CC ** **          CALL      Z,MUL
0300'
0300' 3E C2                       LD    A,0C2H        ;MUL CODE
0302' CD ** **          CALL      PUTeA
0305' C9                          RET
0306'
0306' 3A ** **  _EVDIV: LD        A,(UNDEF)
0309' E6 02                       AND   2
030B' CC ** **          CALL      Z,DIV
030E'
030E' 3E C3                       LD    A,0C3H        ;DIV CODE
0310' CD ** **          CALL      PUTeA
0313' C9                          RET
0314'
0314' 3A ** **  _EVMOD: LD        A,(UNDEF)
0317' E6 02                       AND   2
0319' CC ** **          CALL      Z,MOD
031C'
031C' 3E C7                       LD    A,0C7H        ;MOD CODE
031E' CD ** **          CALL      PUTeA
0321' C9                          RET
0322'
0322' 3A ** **  _EVMINS:LD        A,(UNDEF)
0325' E6 02                       AND   2
0327' CC ** **          CALL      Z,MINS
032A'
032A' 3E C6                       LD    A,0C6H        ;MINUSE CODE
032C' CD ** **          CALL      PUTeA
032F' C9                          RET
0330'
0330' CD ** **  _EVBIN: CALL      BIN
0333' 18 0E                       JR    CHRS1
0335'
0335' CD ** **  _EVHEX: CALL      HEX
0338' 38 09                       JR    C,CHRS1
033A' C9                          RET                 ;if '$' means PC,RET
033B'
033B'
033B' CD ** **  _EVDECI:CALL      DECI
033E' 18 03                       JR    CHRS1
0340'
0340' CD ** **  _EVCHAR:CALL      CHAR
0343' 3E E7             CHRS1:    LD    A,0E7H
0345' CD ** **          CALL      PUTeAHL
0348' C9                          RET
0349'
0349'                   ;
0349'                   ; Evaluate expression only for JR & DJNZ
0349'                   ;
0349' AF                JRVAL::   XOR   A
034A' 32 ** **          LD        (UNDEF),A   ;Clear UNDEF flag
034D' 32 ** **          LD        (JRSEG),A   ;Clear JRSEG
0350' 32 ** **          LD        (CNTeA),A   ;Clear CNTeA
0353'
0353' 3A ** **          LD        A,(pass)
0356' A7                          AND   A
0357' CA ** **          JP        Z,EVAL      ;Check Syntax Error on PASS-1
035A'
035A' CD ** **          CALL      JRVAL10
035D' CD ** **  JRVAL0: CALL      SPCUT
0360' FE 2B                       CP    '+'
0362' 20 08                       JR    NZ,JRVAL1
0364' E5                          PUSH  HL
0365' CD ** **          CALL      JRVAL9
0368' C1                          POP   BC
0369' 09                          ADD   HL,BC
036A' 18 F1                       JR    JRVAL0
036C' FE 2D             JRVAL1:   CP    '-'
036E' C0                          RET   NZ
036F' E5                          PUSH  HL
0370' CD ** **          CALL      JRVAL9
0373' 44                          LD    B,H
0374' 4D                          LD    C,L
0375' E1                          POP   HL
0376' B7                          OR    A
0377' ED 42                       SBC   HL,BC
0379' 18 E2                       JR    JRVAL0
037B'
037B' 13                JRVAL9:   INC   DE
037C' CD ** **  JRVAL10:CALL      SPCUT
037F' FE 2D                       CP    '-'
0381' CA ** **          JP        Z,MINS
0384' FE 24                       CP    '$'
0386' CA ** **          JP        Z,JRHEX
0389' FE 25                       CP    '%'
038B' CA ** **          JP        Z,BIN
038E' CD ** **          CALL      ISDEC
0391' D2 ** **          JP        NC,DECI
0394' CD ** **          CALL      ISQ
0397' CA ** **          JP        Z,CHAR
039A' 21 ** **          LD        HL,ESPTBL
039D' 01 ** **          LD        BC,SPREND-ESPTBL
03A0' ED B1                       CPIR
03A2' CA ** **          JP        Z,ERR9
03A5'
03A5' 3A ** **          LD        A,(UNDEF)
03A8' F6 02                       OR    2
03AA' 32 ** **          LD        (UNDEF),A
03AD'
03AD' CD ** **          CALL      SEALBL
03B0' DA ** **          JP        C,ERR1      ;Undefined Label
03B3'
03B3' A7                          AND   A             ;Uncertained
03B4' CA ** **          JP        Z,ERR3      ;       IllegalLabels
03B7' FE 05                       CP    5             ;EXT
03B9' CA ** **          JP        Z,ERR3      ;       IllegalLabels
03BC'
03BC' 60                          LD    H,B
03BD' 69                          LD    L,C
03BE'
03BE' 47                          LD    B,A
03BF' 3A ** **          LD        A,(JRSEG)
03C2' A7                          AND   A
03C3' 28 04                       JR    Z,JRVAL12
03C5' B8                          CP    B
03C6' C2 ** **          JP        NZ,ERR9             ;Illegal Expression
03C9' 78                JRVAL12:LD      A,B
03CA' 32 ** **          LD        (JRSEG),A
03CD' C9                          RET
03CE'                   ;
03CE'                   ;         print in decimal
03CE'                   ;
03CE' D5                PRDEC::   PUSH  DE
03CF' F5                          PUSH  AF
03D0' AF                          XOR   A
03D1' 32 ** **          LD        (DECWK+5),A
03D4' 01 0A 05          LD        BC,050AH
03D7' 11 ** **          LD        DE,DECWK+5
03DA' CD ** **  PRDEC0: CALL      DIVC
03DD' C6 30                       ADD   A,'0'
03DF' 1B                          DEC   DE
03E0' 12                          LD    (DE),A
03E1' 10 F7                       DJNZ  PRDEC0
03E3' 6B                          LD    L,E
03E4' 62                          LD    H,D
03E5' 01 20 04          LD        BC,0420H
03E8' 3E 30                       LD    A,'0'
03EA' BE                PRDEC1:   CP    (HL)
03EB' 20 04                       JR    NZ,PRDEC2
03ED' 71                          LD    (HL),C
03EE' 23                          INC   HL
03EF' 10 F9                       DJNZ  PRDEC1
03F1' F1                PRDEC2:   POP   AF
03F2' B7                          OR    A
03F3' 28 01                       JR    Z,PRDEC3
03F5' EB                          EX    DE,HL
03F6' CD E5 1F  PRDEC3: CALL      _MSX
03F9' D1                          POP   DE
03FA' C9                          RET
03FB'                   ;
03FB'                   ;         HL = HL * BC
03FB'                   ;
03FB' D5                MUL::     PUSH  DE
03FC' EB                          EX    DE,HL
03FD' 21 00 00          LD        HL,0
0400' 3E 10                       LD    A,16
0402' 29                MUL0:     ADD   HL,HL
0403' EB                          EX    DE,HL
0404' 29                          ADD   HL,HL
0405' EB                          EX    DE,HL
0406' 30 01                       JR    NC,MUL1
0408' 09                          ADD   HL,BC
0409' 3D                MUL1:     DEC   A
040A' 20 F6                       JR    NZ,MUL0
040C' D1                          POP   DE
040D' C9                          RET
040E'                   ;
040E'                   ;         HL/BC=HL...DE
040E'                   ;
040E' D5                DIV:      PUSH  DE
040F' CD ** **          CALL      MOD
0412' EB                          EX    DE,HL
0413' D1                          POP   DE
0414' C9                          RET
0415'                   ;
0415'                   ;         HL/C=HL...A
0415'                   ;
0415' C5                DIVC:     PUSH  BC
0416' AF                          XOR   A
0417' 06 10                       LD    B,16
0419' 29                DIVC0:    ADD   HL,HL
041A' 17                          RLA
041B' 2C                          INC   L
041C' 91                          SUB   C
041D' 30 02                       JR    NC,DIVC1
041F' 2D                          DEC   L
0420' 81                          ADD   A,C
0421' 10 F6             DIVC1:    DJNZ  DIVC0
0423' C1                          POP   BC
0424' C9                          RET
0425'                   ;
0425'                   ;         HL/BC=DE...HL
0425'                   ;
0425' 78                MOD:      LD    A,B
0426' B1                          OR    C
0427' 28 30                       JR    Z,ERR9
0429' EB                          EX    DE,HL
042A' 21 00 00          LD        HL,0
042D' 3E 10                       LD    A,16
042F' EB                MOD0:     EX    DE,HL
0430' 29                          ADD   HL,HL
0431' EB                          EX    DE,HL
0432' ED 6A                       ADC   HL,HL
0434' 1C                          INC   E
0435' ED 42                       SBC   HL,BC
0437' 30 02                       JR    NC,MOD1
0439' 09                          ADD   HL,BC
043A' 1D                          DEC   E
043B' 3D                MOD1:     DEC   A
043C' 20 F1                       JR    NZ,MOD0
043E' C9                          RET
043F'
043F'
043F'                   ;
043F'                   ;         error
043F'                   ;
043F' AF                ERR0::    XOR   A
0440' 01                          DEFB  01H
0441' 3E 01             ERR1::    LD    A,1
0443' 01                          DEFB  01H
0444' 3E 02             ERR2::    LD    A,2
0446' 01                          DEFB  01H
0447' 3E 03             ERR3::    LD    A,3
0449' 01                          DEFB  01H
044A' 3E 04             ERR4::    LD    A,4
044C' 01                          DEFB  01H
044D' 3E 05             ERR5::    LD    A,5
044F' 01                          DEFB  01H
0450' 3E 06             ERR6::    LD    A,6
0452' 01                          DEFB  01H
0453' 3E 07             ERR7::    LD    A,7
0455' 01                          DEFB  01H
0456' 3E 08             ERR8::    LD    A,8
0458' 01                          DEFB  01H
0459' 3E 09             ERR9::    LD    A,9
045B' 01                          DEFB  01H
045C' 3E 0A             ERR10::   LD    A,10
045E' 01                          DEFB  01H
045F' 3E 0B             ERR11::   LD    A,11
0461' 01                          DEFB  01H
0462' 3E 0C             ERR12::   LD    A,12
0464' 01                          DEFB  01H
0465' 3E 0D             ERR13::   LD    A,13
0467' 01                          DEFB  01H
0468' 3E 0E             ERR14::   LD    A,14
046A' 01                          DEFB  01H
046B' 3E 0F             ERR15::   LD    A,15
046D' 01                          DEFB  01H
046E' 3E 10             ERR16::   LD    A,16
0470' 32 ** **          LD        (ERRNUM),A
0473' ED 53 ** **       ERROR:    LD    (ERRPTR),DE
0477' 87                          ADD   A,A
0478' 21 ** **          LD        HL,ERRTBL
047B' 4F                          LD    C,A
047C' 06 00                       LD    B,0
047E' 09                          ADD   HL,BC
047F' 5E                          LD    E,(HL)
0480' 23                          INC   HL
0481' 56                          LD    D,(HL)
0482' CD E5 1F          CALL      _MSX
0485' 11 ** **          LD        DE,ERRMSG
0488' CD E5 1F          CALL      _MSX
048B' CD ** **          CALL      PRTXT
048E' 3A ** **          LD        A,(ERRPOS)
0491' 47                          LD    B,A
0492' CD DF 1F          CALL      _TAB
0495' 3E 5E                       LD    A,'^'
0497' CD F4 1F          CALL      _PRINT
049A' CD EE 1F          CALL      _LTNL
049D' 2A ** **          LD        HL,(ERRCNT)
04A0' 23                          INC   HL
04A1' 22 ** **          LD        (ERRCNT),HL
04A4' ED 5B ** **                 LD    DE,(LINPTR)
04A8' CD ** **          CALL      NXLIN
04AB' ED 7B ** **                 LD    SP,(SPBUF)
04AF' B7                          OR    A
04B0' C9                          RET
04B1'
04B1' D5                PRTXT:    PUSH  DE
04B2' CD F1 1F          CALL      _PRNTS
04B5' 2A ** **          LD        HL,(LINNO)
04B8' AF                          XOR   A
04B9' CD ** **          CALL      PRDEC
04BC' CD F1 1F          CALL      _PRNTS
04BF' CD F1 1F          CALL      _PRNTS
04C2' ED 5B ** **                 LD    DE,(LINPTR)
04C6' 2A ** **  PRTXT0: LD        HL,(ERRPTR)
04C9' B7                          OR    A
04CA' ED 52                       SBC   HL,DE
04CC' 20 07                       JR    NZ,PRTXT1
04CE' 2A 7A 1F          LD        HL,(_PRCNT)
04D1' 7E                          LD    A,(HL)
04D2' 32 ** **          LD        (ERRPOS),A
04D5' 1A                PRTXT1:   LD    A,(DE)
04D6' 13                          INC   DE
04D7' FE 0D                       CP    CR
04D9' 28 0A                       JR    Z,PRTXT3
04DB' FE 09                       CP    TAB
04DD' C4 F4 1F          CALL      NZ,_PRINT
04E0' CC ** **          CALL      Z,PRTTAB
04E3' 18 E1                       JR    PRTXT0
04E5'
04E5' D1                PRTXT3:   POP   DE
04E6' CD EE 1F          CALL      _LTNL
04E9' C9                          RET
04EA'
04EA' CD F1 1F  PRTTAB: CALL      _PRNTS
04ED' 2A 7A 1F          LD        HL,(_PRCNT)
04F0' 7E                          LD    A,(HL)
04F1' E6 07                       AND   7
04F3' 20 F5                       JR    NZ,PRTTAB
04F5' C9                          RET
04F6'
04F6'
04F6'                   ; POKE with Increment HL
04F6' CD 9A 1F  POKE_I::CALL      _POKE
04F9' 23                          INC   HL
04FA' C9                          RET
04FB'
04FB'                   ; POKE BC
04FB'                   POKE_BC::
04FB' F5                          PUSH  AF
04FC' 79                          LD    A,C
04FD' CD 9A 1F          CALL      _POKE
0500' 23                          INC   HL
0501' 78                          LD    A,B
0502' CD 9A 1F          CALL      _POKE
0505' 23                          INC   HL
0506' F1                          POP   AF
0507' C9                          RET
0508'
0508'
0508'                   ; PEEK with Increment HL
0508' CD 94 1F  PEEK_I::CALL      _PEEK
050B' 23                          INC   HL
050C' C9                          RET
```

▶愛する X68000を残し，Oh!X を捨て，私は受験戦争へ突入するのであった。

田端　秀章（17）北海道

```
050D'
050D'                   ; PEEK BC
050D'                   PEEK_BC::
050D' F5                        PUSH   AF
050E' CD 94 1F          CALL    _PEEK
0511' 4F                        LD     C,A
0512' 23                        INC    HL
0513' CD 94 1F          CALL    _PEEK
0516' 47                        LD     B,A
0517' 23                        INC    HL
0518' F1                        POP    AF
0519' C9                        RET
051A'
051A'
051A'                   ;
051A'                   ;       error messages
051A'                   ;
051A' ** ** ** **       ERRTBL: DEFW   ERMS00,ERMS01
051E' ** ** ** **               DEFW   ERMS02,ERMS03
0522' ** ** ** **               DEFW   ERMS04,ERMS05
0526' ** ** ** **               DEFW   ERMS06,ERMS07
052A' ** ** ** **               DEFW   ERMS08,ERMS09
052E' ** ** ** **               DEFW   ERMS10,ERMS11
0532' ** ** ** **               DEFW   ERMS12,ERMS13
0536' ** ** ** **               DEFW   ERMS14,ERMS15
053A' ** **                     DEFW   ERMS16
053C'                   ;
053C' 53 79 6E 74 61    ERMS00: DEFB   'Syntax',0
0541' 78 00
0543' 55 6E 64 65 66    ERMS01: DEFB   'Undefined Label',0
0548' 69 6E 65 64 20
054D' 4C 61 62 65 6C
0552' 00
0553' 52 65 64 65 66    ERMS02: DEFB   'Redefinition',0
0558' 69 6E 69 74 69
055D' 6F 6E 00
0560' 49 6C 6C 65 67    ERMS03: DEFB   'Illegal Label',0
0565' 61 6C 20 4C 61
056A' 62 65 6C 00
056E' 49 6C 6C 65 67    ERMS04: DEFB   'Illegal Opcode',0
0573' 61 6C 20 4F 70
0578' 63 6F 64 65 00
057D'
057D' 49 6C 6C 65 67    ERMS05: DEFB   'Illegal Operand',0
0582' 61 6C 20 4F 70
0587' 65 72 61 6E 64
058C' 00
058D' 54 6F 6F 20 4D    ERMS06: DEFB   'Too Many Labels',0
0592' 61 6E 79 20 4C
0597' 61 62 65 6C 73
059C' 00
059D' 4D 69 73 73 69    ERMS07: DEFB   'Missing Label',0
05A2' 6E 67 20 4C 61
05A7' 62 65 6C 00
05AB' 54 6F 6F 20 46    ERMS08: DEFB   'Too Far',0
05B0' 61 72 00
05B3' 49 6C 6C 65 67    ERMS09: DEFB   'Illegal Expression',0
05B8' 61 6C 20 45 78
05BD' 70 72 65 73 73
05C2' 69 6F 6E 00
05C6' 4D 69 73 73 69    ERMS10: DEFB   'Missing [)]',0
05CB' 6E 67 20 5B 29
05D0' 5D 00
05D2' 4D 69 73 73 69    ERMS11: DEFB   'Missing [.]',0
05D7' 6E 67 20 5B 2E
05DC' 5D 00
05DE' 4D 69 73 73 69    ERMS12: DEFB   'Missing Quote',0
05E3' 6E 67 20 51 75
05E8' 6F 74 65 00
05EC' 00                ERMS13: DEFB   0
05ED' 49 6C 6C 65 67    ERMS14: DEFB   'Illegal ORG',0
05F2' 61 6C 20 4F 52
05F7' 47 00
05F9' 56 61 6C 75 65    ERMS15: DEFB   'Value',0
05FE' 00
05FF' 52 65 6C 6F 63    ERMS16: DEFB   'Relocation',0
0604' 61 74 69 6F 6E
0609' 00
060A' 20 45 72 72 6F    ERRMSG: DEFB   ' Error ',CR,0
060F' 72 20 0D 00
0613'
0613'
0613'
0613'                   ;
0613'                   ;       opecode table
0613'                   ;
0613' ** ** ** **       C0TBL:: DEFW   ATBL,BTBL
0617' ** ** ** **               DEFW   CTBL,DTBL
061B' ** ** ** **               DEFW   ETBL,FTBL
061F' ** ** ** **               DEFW   GTBL,HTBL
0623' ** ** ** **               DEFW   ITBL,JTBL
0627' ** ** ** **               DEFW   KTBL,LTBL
062B' ** ** ** **               DEFW   MTBL,NTBL
062F' ** ** ** **               DEFW   OTBL,PTBL
0633' ** ** ** **               DEFW   QTBL,RTBL
0637' ** ** ** **               DEFW   STBL,TTBL
063B' ** ** ** **               DEFW   UTBL,VTBL
063F' ** ** ** **               DEFW   WTBL,XTBL
0643'                   ;
0080                    n       EQU    80H
0643'                   ;
0643' 44 44 B0  ATBL:   DEFB    'DD',  30H+n
0646' 4E 44 B1          DEFB    'ND',  31H+n
0649' 44 43 B2          DEFB    'DC',  32H+n
064C' 00                        DEFB   0
064D' 49 54 B3  BTBL:   DEFB    'IT',  33H+n
0650' 00                        DEFB   0
0651' 50 B4             CTBL:   DEFB   'P',   34H+n
0653' 41 4C 4C B5               DEFB   'ALL', 35H+n
0657' 43 46 80          DEFB    'CF',  00H+n
065A' 50 4C 81          DEFB    'PL',  01H+n
065D' 50 49 90          DEFB    'PI',  10H+n
0660' 50 44 91          DEFB    'PD',  11H+n
0663' 50 49 52 92               DEFB   'PIR', 12H+n
0667' 50 44 52 93               DEFB   'PDR', 13H+n
066B' 53 45 47 DA               DEFB   'SEG', 5AH+n
066F' 4F 4D 4D 4F 4E            DEFB   'OMMON',5CH+n
0674' DC
0675' 00                        DEFB   0
0676' 45 43 B6  DTBL:   DEFB    'EC',  36H+n
0679' 4A 4E 5A B7               DEFB   'JNZ', 37H+n
067D' 45 46 42 B8               DEFB   'EFB', 38H+n
0681' 45 46 57 B9               DEFB   'EFW', 39H+n
0685' 45 46 53 BA               DEFB   'EFS', 3AH+n
0689' 42 B8                     DEFB   'B',   38H+n
068B' 57 B9                     DEFB   'W',   39H+n
068D' 53 BA                     DEFB   'S',   3AH+n
068F' 41 41 83          DEFB    'AA',  03H+n
0692' 49 82                     DEFB   'I',   02H+n
0694' 4D BB                     DEFB   'M',   3BH+n
0696' 45 46 4D BB               DEFB   'EFM', 3BH+n
069A' 53 45 47 DB               DEFB   'SEG', 5BH+n
069E' 00                        DEFB   0
069F' 58 BC             ETBL:   DEFB   'X',   3CH+n
06A1' 58 58 84          DEFB    'XX',  04H+n
06A4' 51 55 BD          DEFB    'QU',  3DH+n
06A7' 49 85                     DEFB   'I',   05H+n
06A9' 58 54 DF          DEFB    'XT',  5FH+n
06AC' 4E 44 E2          DEFB    'ND',  62H+n
06AF' 00                        DEFB   0
06B0' 41 4C 54 86       HTBL:   DEFB   'ALT', 06H+n
06B4' 00                        DEFB   0
06B5' 4E 43 BE  ITBL:   DEFB    'NC',  3EH+n
06B8' 4E BF                     DEFB   'N',   3FH+n
06BA' 4E 49 94          DEFB    'NI',  14H+n
06BD' 4E 44 95          DEFB    'ND',  15H+n
06C0' 4E 49 52 96               DEFB   'NIR', 16H+n
06C4' 4E 44 52 97               DEFB   'NDR', 17H+n
06C8' 4D C0                     DEFB   'M',   40H+n
06CA' 4E 43 4C 55 44            DEFB   'NCLUDE',61H+n
06CF' 45 E1
06D1' 00                        DEFB   0
06D2' 52 C1             JTBL:   DEFB   'R',   41H+n
06D4' 50 C2                     DEFB   'P',   42H+n
06D6' 00                        DEFB   0
06D7' 44 C3             LTBL:   DEFB   'D',   43H+n
06D9' 44 49 52 98               DEFB   'DIR', 18H+n
06DD' 44 49 99          DEFB    'DI',  19H+n
06E0' 44 44 52 9A               DEFB   'DDR', 1AH+n
06E4' 44 44 9B          DEFB    'DD',  1BH+n
06E7' 00                        DEFB   0
06E8' 45 47 9C  NTBL:   DEFB    'EG',  1CH+n
06EB' 4F 50 87          DEFB    'OP',  07H+n
06EE' 00                        DEFB   0
06EF' 52 C4             OTBL:   DEFB   'R',   44H+n
06F1' 55 54 C5          DEFB    'UT',  45H+n
06F4' 55 54 49 9D               DEFB   'UTI', 1DH+n
06F8' 54 49 52 9E               DEFB   'TIR', 1EH+n
06FC' 55 54 44 9F               DEFB   'UTD', 1FH+n
0700' 54 44 52 A0               DEFB   'TDR', 20H+n
0704' 52 47 C6          DEFB    'RG',  46H+n
0707' 00                        DEFB   0
0708' 4F 50 C8  PTBL:   DEFB    'OP',  48H+n
070B' 55 53 48 C9               DEFB   'USH', 49H+n
070F' 55 42 4C 49 43            DEFB   'UBLIC',60H+n
0714' E0
0715' 41 47 45 E3               DEFB   'AGE', 63H+n
0719' 00                        DEFB   0
071A' 45 54 CA  RTBL:   DEFB    'ET',  4AH+n
071D' 4C 41 88          DEFB    'LA',  08H+n
0720' 52 41 89          DEFB    'RA',  09H+n
0723' 4C 43 41 8A               DEFB   'LCA', 0AH+n
0727' 52 43 41 8B               DEFB   'RCA', 0BH+n
072B' 4C CB                     DEFB   'L',   4BH+n
072D' 52 CC                     DEFB   'R',   4CH+n
072F' 4C 43 CD          DEFB    'LC',  4DH+n
0732' 52 43 CE          DEFB    'RC',  4EH+n
0735' 4C 44 A1          DEFB    'LD',  21H+n
0738' 52 44 A2          DEFB    'RD',  22H+n
073B' 45 53 CF          DEFB    'ES',  4FH+n
073E' 53 54 D0          DEFB    'ST',  50H+n
0741' 45 54 49 A3               DEFB   'ETI', 23H+n
0745' 45 54 4E A4               DEFB   'ETN', 24H+n
0749' 00                        DEFB   0
074A' 42 43 D1  STBL:   DEFB    'BC',  51H+n
074D' 55 42 D2          DEFB    'UB',  52H+n
0750' 43 46 8C          DEFB    'CF',  0CH+n
0753' 45 54 D3          DEFB    'ET',  53H+n
0756' 4C 41 D4          DEFB    'LA',  54H+n
0759' 52 41 D5          DEFB    'RA',  55H+n
075C' 52 4C D6          DEFB    'RL',  56H+n
075F' 00                        DEFB   0
0760' 49 54 4C 45 E4    TTBL:   DEFB   'ITLE',     64H+n
0765'
0765' 00                        DEFB   0
0766' 4F 52 D7  XTBL:   DEFB    'OR',  57H+n
0769' 00                        DEFB   0
076A' 4C 49 53 54 D8    @TBL::  DEFB   'LIST',        58H+n
076F'
076F' 4E 4C 49 53 54            DEFB   'NLIST',    59H+n
0774' D9
0775' 50 48 41 53 45            DEFB   'PHASE',    5DH+n
077A' DD
077B' 44 45 50 48 41            DEFB   'DEPHASE',  5EH+n
0780' 53 45 DE
0783' 4B 41 4E 4A 49            DEFB   'KANJI'
0788'
0788' 4E 4B 41 4E 4A            DEFB   'NKANJI'
078D' 49
078E'                   FTBL:
078E'                   GTBL:
078E'                   KTBL:
078E'                   MTBL:
078E'                   QTBL:
078E'                   UTBL:
078E'                   VTBL:
078E'                   WTBL:
078E' 00                        DEFB   0
078F'                   ;
078F' 3F                C1TBL:: CCF                  ;00
0790' 2F                        CPL                  ;01
0791' F3                        DI                   ;02
0792' 27                        DAA                  ;03
0793' D9                        EXX                  ;04
0794' FB                        EI                   ;05
0795' 76                        HALT                 ;06
0796' 00                        NOP                  ;07
0797' 17                        RLA                  ;08
0798' 1F                        RRA                  ;09
0799' 07                        RLCA                 ;0A
079A' 0F                        RRCA                 ;0B
079B' 37                        SCF                  ;0C
079C'                   ;
079C' ED A1             C2TBL:: CPI                  ;10
079E' ED A9                     CPD                  ;11
07A0' ED B1                     CPIR                 ;12
07A2' ED B9                     CPDR                 ;13
07A4' ED A2                     INI                  ;14
07A6' ED AA                     IND                  ;15
07A8' ED B2                     INIR                 ;16
07AA' ED BA                     INDR                 ;17
07AC' ED B0                     LDIR                 ;18
07AE' ED A0                     LDI                  ;19
07B0' ED B8                     LDDR                 ;1A
07B2' ED A8                     LDD                  ;1B
07B4' ED 44                     NEG                  ;1C
07B6' ED A3                     OUTI                 ;1D
07B8' ED B3                     OTIR                 ;1E
07BA' ED AB                     OUTD                 ;1F
07BC' ED BB                     OTDR                 ;20
07BE' ED 6F                     RLD                  ;21
07C0' ED 67                     RRD                  ;22
07C2' ED 4D                     RETI                 ;23
07C4' ED 45                     RETN                 ;24
07C6'
07C6'
07C6'                   ;
07C6' 42 43 00  REGTBL::DEFB    'BC',0        :00
07C9' 44 45 00          DEFB    'DE',0      :01
07CC' 48 4C 00          DEFB    'HL',0      :02
07CF' 53 50 00          DEFB    'SP',0      :03
07D2' 41 46 00          DEFB    'AF',0      :04
07D5' 42 00             RG8TBL::DEFB   'B',0          :05     00
07D7' 43 00                     DEFB   'C',0        ;0601
07D9' 44 00                     DEFB   'D',0        ;0702
07DB' 45 00                     DEFB   'E',0        ;0803
07DD' 48 00                     DEFB   'H',0        ;0904
07DF' 4C 00                     DEFB   'L',0        ;0A05
07E1' 28 48 4C 29 00            DEFB   '(HL)',0     ;0B06
07E6'
07E6' 41 00                     DEFB   'A',0        ;0C07
07E8' 49 58 00          DEFB    'IX',0      :0D
07EB' 49 59 00          DEFB    'IY',0      :0E
07EE' 28 49 58 00               DEFB   '(IX',0      ;0F
07F2' 28 49 59 00               DEFB   '(IY',0      ;10
07F6' 00                        DEFB   0
07F7'                   ;
07F7' 4E 5A 00  CNDTBL::DEFB    'NZ',0        ;00
07FA' 5A 00                     DEFB   'Z',0        ;01
07FC' 4E 43 00          DEFB    'NC',0      ;02
07FF' 43 00                     DEFB   'C',0        ;03
0801' 50 4F 00          DEFB    'PO',0      :04
0804' 50 45 00          DEFB    'PE',0      :05
0807' 50 00                     DEFB   'P',0        ;06
0809' 4D 00                     DEFB   'M',0        ;07
080B' 00                        DEFB   0
080C'
080C' 28 44 45 29 2C    [|]TBL:: DEFB  '(DE),A',0
0811' 41 00
0813' 12                        LD     (DE),A
0814' 00                        NOP
0815' 28 42 43 29 2C            DEFB   '(BC),A',0
081A' 41 00
081C' 02                        LD     (BC),A
081D' 00                        NOP
081E' 52 2C 41 00               DEFB   'R,A',0
0822' ED 4F                     LD     R,A
0824' 49 2C 41 00               DEFB   'I,A',0
0828' ED 47                     LD     I,A
082A' 00                        DEFB   0
082B'                   ;
082B' 44 45 29 00       A[|]TBL::DEFB  'DE)',0
082F' 1A                        LD     A,(DE)
0830' 00                        NOP
0831' 42 43 29 00               DEFB   'BC)',0
0835' 0A                        LD     A,(BC)
0836' 00                        NOP
0837' 00                        DEFB   0
0838'                   ;
0838' 44 45 2C 48 4C    EXTBL:: DEFB   'DE,HL',0
083D' 00
083E' EB                        EX     DE,HL
083F' 00                        NOP
0840' 41 46 2C 41 46            DEFB   "AF,AF'",0
0845' 27 00
0847' 08                        EX     AF,AF'
0848' 00                        NOP
0849' 28 53 50 29 2C            DEFB   '(SP),HL',0
084E' 48 4C 00
0851' E3                        EX     (SP),HL
0852' 00                        NOP
0853' 28 53 50 29 2C            DEFB   '(SP),IX',0
0858' 49 58 00
085B' DD E3                     EX     (SP),IX
085D' 28 53 50 29 2C            DEFB   '(SP),IY',0
0862' 49 59 00
0865' FD E3                     EX     (SP),IY
0867' 00                        DEFB   0
0868'                   ;
0868' 28 48 4C 29 00    JPTBL:: DEFB   '(HL)',0
086D'
086D' E9                        JP     (HL)
086E' 00                        NOP
086F' 28 49 58 29 00            DEFB   '(IX)',0
0874'
0874' DD E9                     JP     (IX)
0876' 28 49 59 29 00            DEFB   '(IY)',0
087B'
087B' FD E9                     JP     (IY)
087D' 00                        DEFB   0
087E'                   ;
087E' 09 20             LSPTBL::DEFB   TAB,SPC
0880' 0D 3B 3A 29 2C    ESPTBL: DEFB   CR,";:),+-*/%",'"',"'"
0885' 2B 2D 2A 2F 25
088A' 22 27
088C'                   SPREND::
088C'
088C'                   ;
088C'                   ; Works
088C'                   ;
088C'                           DSEG
0000"                   ERRNUM::DEFS   2
0002"                   ERRPOS: DEFS   1
0003"                   ERRPTR: DEFS   2
0005" 00 00 00 00 00    DECWK:  DEFW   0,0,0
000A" 00
000B"
000B"
```

リスト6　WZDソースリスト5

```
S0000'                  ;**********************************
 0000'                  ; File Access Routine For WZD
 0000'                  ;         Programed by T.Ishigami
 0000'                  ;          '89 Jun.2nd
 0000'                  ;
 0000'                  ;**********************************
 0000'
 0000'                          CSEG
 0000'                  ;
 0000'                  ; Externals
 0000'                  ;
                                EXT    wkso
                                EXT    wkre
                                EXT    wkpr
                                EXT    flvl
                                EXT    MUL
 0000'
 0000'                C  INCLUDE  WZD.DEF
 0000'                C ; Header File For WZD
 0000'                C ;   CSEG 3000H-
 0000'                C ;   DSEG 6000H-
 0000'                C ;
 0000'                C
 B800                 C RDBUF     EQU 0B800H
 BC00                 C WRBUF1    EQU 0BC00H
 BD00                 C WRBUF2    EQU 0BD00H
 0000'                C
 0000'                C  INCLUDE  SOS.DEF
 1FFA                 C _HOT      EQU    1FFAH
 1FF4                 C _PRINT    EQU    1FF4H
 1FF1                 C _PRNTS    EQU    1FF1H
 1FEE                 C _LTNL     EQU    1FEEH
 1FEB                 C _NL       EQU    1FEBH
 1FE5                 C _MSX      EQU    1FE5H
 1FDF                 C _TAB      EQU    1FDFH
 1FD3                 C _GETL     EQU    1FD3H
 1FC7                 C _PAUSE    EQU    1FC7H
 1FC1                 C _PRTHX    EQU    1FC1H
 1FBE                 C _PRTHL    EQU    1FBEH
 1FB8                 C _HEX      EQU    1FB8H
 1FA3                 C _FILE     EQU    1FA3H
 1FAF                 C _WOPEN    EQU    1FAFH
 1F9A                 C _POKE     EQU    1F9AH
 1F94                 C _PEEK     EQU    1F94H
 2009                 C _ROPEN    EQU    2009H
 2015                 C _KILL     EQU    2015H
 2012                 C _NAME     EQU    2012H
 2033                 C _ERROR    EQU    2033H
 0000'                C
 2000                 C _DRDSB    EQU    2000H
 2003                 C _DWTSB    EQU    2003H
 0000'                C
 1F7A                 C _PRCNT    EQU    1F7AH
 1F76                 C _KBFAD    EQU    1F76H
 1F74                 C _IBFAD    EQU    1F74H
 1F72                 C _SIZE     EQU    1F72H
 1F6A                 C _MEMAX    EQU    1F6AH
 1F64                 C _DTBUF    EQU    1F64H
 1F62                 C _FATBF    EQU    1F62H
 1F60                 C _DIRPS    EQU    1F60H
 1F5E                 C _FATPOS   EQU    1F5EH
 1F5D                 C _DSK      EQU    1F5DH
 0000'                C
 0000'
 0000'
 0000'                  ;====================
 0000'                  ; FILE OPEN FOR READ
 0000'                  ;====================
 0000' 22 ** **  RDOPEN::LD         (FCB_ADR),HL
 0003'
 0003' CD A3 1F          CALL       _FILE
 0006' D8                           RET    C
 0007' CD ** **          CALL       ROPEN
 000A' D8                           RET    C
 000B'                              ;
 000B' 2A 74 1F          LD         HL,(_IBFAD)
 000E' 11 ** **          LD         DE,FILE_BF
 0011' 01 20 00          LD         BC,20H
 0014' ED B0                        LDIR
 0016' 3A 5D 1F          LD         A,(_DSK)
 0019' 32 ** **          LD         (FLDSK),A
 001C'
 001C' AF                           XOR    A
 001D' 32 ** **          LD         (LST_DSK),A  ;NEVER BEING THE SAME
 0020'
 0020' CD ** **          CALL       RDFAT
 0023' D8                           RET    C
 0024'
 0024' 06 10                        LD     B,10H
 0026' 0E 00                        LD     C,0           ; C <= TOTAL NUMBER OF CLUSTE
RS
 0028' 3A ** **          LD         A,(FSTCLST)
 002B' 11 ** **          LD         DE,TBLCLST
 002E'
 002E' 12                RDOPN4:    LD     (DE),A
 002F' 13                           INC    DE
 0030' FE 7F                        CP     7FH
 0032' 30 0F                        JR     NC,RDOPN5
 0034' 2A 62 1F          LD         HL,(_FATBF)
 0037' 85                           ADD    A,L
 0038' 6F                           LD     L,A
 0039' 30 01                        JR     NC,RDOPN2
 003B' 24                           INC    H
 003C' 7E                RDOPN2:    LD     A,(HL)        ;HL = (_FATBF) + (FSTCLST)
 003D' 05                           DEC    B
 003E' 28 24                        JR     Z,RDOPN3      ; MORE THAN 16 CLUSTER
 0040' 0C                           INC    C
 0041' 18 EB                        JR     RDOPN4
 0043'
 0043' 0D                RDOPN5:    DEC    C             ;LAST CLUSTER IS DUMMY
 0044' F5                           PUSH   AF
 0045' 79                           LD     A,C
 0046' 87                           ADD    A,A
 0047' 87                           ADD    A,A
```

▶18歳になったのに，車の免許がとれない。なぜなら私は予備校生。18になるのを楽しみにしていたのに……。彼女は，「私が免許をとったら乗せてあげるわよ」といっているが……。俺はまだ死にたくない。

増永　哲也（18）福井県

```
0048' 87                      ADD    A,A
0049' 87                      ADD    A,A
004A' 4F                      LD     C,A          ;C = C * 16
004B' F1                      POP    AF
004C' D6 80                   SUB    80H          ;RC is 0 origin. So it isn't
7Fh but 80H.
004E' 81                      ADD    A,C
004F' 32 ** **          LD    (RC),A        ;RC = Total number of Records
0052'
0052' AF                      XOR    A
0053' 32 ** **          LD    (FLPNT),A
0056'
0056' 01 38 00          LD    BC,INF_SIZE
0059' ED 5B ** **             LD     DE,(FCB_ADR)
005D' 21 ** **          LD    HL,FILE_BF
0060' ED B0                   LDIR
0062'
0062' A7                      AND    A
0063' C9                      RET
0064'
0064' 3E 07             RDOPN3: LD   A,7          ;Bad AllocationError
0066' 37                      SCF
0067' C9                      RET
0068'
0068'
0068' 3A 5D 1F  ROPEN:  LD    A,(_DSK)
006B' CD ** **          CALL  DEVCHK
006E' D8                      RET    C            ;Bad File descripter
006F' CD ** **          CALL  FCBSCH
0072' D8                      RET    C
0073' 3E 08                   LD     A,8          ;File Not Fonnd
0075' 37                      SCF
0076' C0                      RET    NZ
0077' E5                      PUSH   HL
0078' ED 5B 74 1F             LD     DE,(_IBFAD)
007C' 01 20 00          LD    BC,20H
007F' ED B0                   LDIR
0081' E1                      POP    HL
0082' 7E                      LD     A,(HL)
0083' CD ** **          CALL  FMCHK
0086' C9                      RET
0087'
0087' FE 41             DEVCHK: CP   'A'
0089' 38 04                   JR     C,DEVCH1
008B' FE 45                   CP     'D'+1
008D' 3F                      CCF
008E' D0                      RET    NC
008F'
008F' 3E 03             DEVCH1: LD   A,3   ;Bad File descripter
0091' C9                      RET
0092'
0092'                   ;
0092'                   ; FCB SEARCH
0092'                   ;
0092' 0E 10             FCBSCH: LD   C,16         ;Directory Lengh
0094' ED 5B 60 1F             LD     DE,(_DIRPS)  ;Directory start
0098' ED 53 ** **       FCBSC1: LD   (DEBUF),DE
009C' 2A 64 1F          LD    HL,(_DTBUF)
009F' 3E 01                   LD     A,1
00A1' CD 00 20          CALL  _DRDSB
00A4' D8                      RET    C
00A5' 06 08                   LD     B,8
00A7' 22 ** **  FCBSC2: LD    (HLBUF),HL
00AA' 7E                      LD     A,(HL)
00AB' FE FF                   CP     0FFH
00AD' 28 1A                   JR     Z,FCBSC4
00AF' B7                      OR     A
00B0' 28 0B                   JR     Z,FCBSC3
00B2' D5                      PUSH   DE
00B3' ED 5B 74 1F             LD     DE,(_IBFAD)
00B7' CD ** **          CALL  FCOMP
00BA' D1                      POP    DE
00BB' 28 0D                   JR     Z,FCBSC5
00BD' D5                FCBSC3: PUSH DE
00BE' 11 20 00          LD    DE,32
00C1' 19                      ADD    HL,DE
00C2' D1                      POP    DE
00C3' 10 E2                   DJNZ   FCBSC2
00C5' 13                      INC    DE
00C6' 0D                      DEC    C
00C7' 20 CF                   JR     NZ,FCBSC1
00C9'
00C9' 3E                FCBSC4: DB   3EH   ; Z = 0
00CA' AF                FCBSC5: XOR  A     ; Z = 1
00CB' B7                      OR     A
00CC' C9                      RET
00CD'
00CD'
00CD'                   ; File Name Compare
00CD' C5                FCOMP:  PUSH BC
00CE' E5                      PUSH   HL
00CF' 06 10                   LD     B,16  ;Directory lengh
00D1' 13                FCOMP1: INC  DE
00D2' 23                      INC    HL
00D3' 1A                      LD     A,(DE)
00D4' BE                      CP     (HL)
00D5' 20 02                   JR     NZ,FCOMP2
00D7' 10 F8                   DJNZ   FCOMP1
00D9' E1                FCOMP2: POP  HL
00DA' C1                      POP    BC
00DB' C9                      RET
00DC'
00DC'                   ; FILE MODE CHECK
00DC' E5                FMCHK:  PUSH HL
00DD' E6 87                   AND    87H          ;1000&0111B
00DF' 21 1F 29          LD    HL,291FH     ;%FTYPE
00E2' BE                      CP     (HL)
00E3' E1                      POP    HL
00E4' C8                      RET    Z
00E5' 3E 06                   LD     A,6          ;Bad File Mode
00E7' 37                      SCF
00E8' C9                      RET
00E9'
00E9'
00E9'                   ;=====================
00E9'                   ; FILE OPEN FOR WRITE
00E9'                   ;=====================
00E9' 22 ** **  WROPEN::LD    (FCB_ADR),HL
00EC' CD A3 1F          CALL  _FILE
00EF' D8                      RET    C
00F0' CD AF 1F          CALL  _WOPEN
00F3' D8                      RET    C
00F4'
00F4' 01 20 00          LD    BC,20H
00F7' 11 ** **          LD    DE,FILE_BF
00FA' 2A 74 1F          LD    HL,(_IBFAD)
00FD' ED B0                   LDIR
00FF'
00FF' 3A 5D 1F          LD    A,(_DSK)
0102' 32 ** **          LD    (FLDSK),A
0105'
0105' 2A E1 27          LD    HL,(27E1H)
0108' 22 ** **          LD    (HLBUF),HL
010B' 2A DF 27          LD    HL,(27DFH)
010E' 22 ** **          LD    (DEBUF),HL
0111'
0111'                         ;CALL RDFAT         ;FAT has already loaded in _W
OPEN
0111'
0111' CD ** **          CALL  FCGET
0114' D8                      RET    C
0115'
0115' 16 00                   LD     D,0
0117' 5F                      LD     E,A
0118' 2A 62 1F          LD    HL,(_FATBF)
011B' 19                      ADD    HL,DE
011C' 36 8F                   LD     (HL),8FH     ;Dummy Data
011E'
011E' 32 ** **          LD    (FSTCLST),A
0121' 32 ** **          LD    (TBLCLST),A
0124' 3E 80                   LD     A,80H
0126' 32 ** **          LD    (TBLCLST+1),A
0129'
0129' CD ** **          CALL  WRFAT
012C' D8                      RET    C
012D'
012D' AF                      XOR    A
012E' 32 ** **          LD    (RC),A
0131' 32 ** **          LD    (LST_DSK),A  ;Never being then same
0134' 32 ** **          LD    (FLPNT),A
0137'
0137' 21 00 00          LD    HL,0
013A' 22 ** **          LD    (FLSIZE),HL
013D'
013D' 01 20 00          LD    BC,20H
0140' ED 5B ** **             LD     DE,(HLBUF)
0144' 2A 74 1F          LD    HL,(_IBFAD)
0147' ED B0                   LDIR
0149'
0149' 3E 01                   LD     A,1
014B' ED 5B ** **             LD     DE,(DEBUF)
014F' 2A 64 1F          LD    HL,(_DTBUF)
0152' CD 03 20          CALL  _DWTSB
0155' D8                      RET    C
0156'
0156' 01 38 00          LD    BC,INF_SIZE
0159' ED 5B ** **             LD     DE,(FCB_ADR)
015D' 21 ** **          LD    HL,FILE_BF
0160' ED B0                   LDIR
0162'
0162' B7                      OR     A            ;RCF
0163' C9                      RET
0164'
0164'                   ;==============
0164'                   ; FILE CLOSE
0164'                   ;==============
0164' 22 ** ** CLOSE:: LD     (FCB_ADR),HL
0167' 11 ** **          LD    DE,FILE_BF
016A' 01 38 00          LD    BC,INF_SIZE
016D' ED B0                   LDIR
016F'
016F' 3A ** **          LD    A,(FLDSK)
0172' 32 5D 1F          LD    (_DSK),A
0175'
0175' 2A ** **          LD    HL,(FLSIZE)
0178' 2C                      INC    L
0179' 2D                      DEC    L
017A' C4 ** **          CALL  NZ,WRITE1
017D'
017D' 3A ** **          LD    A,(RC)
0180' 2A ** **          LD    HL,(FLSIZE)
0183' 2C                      INC    L
0184' 2D                      DEC    L
0185' 20 01                   JR     NZ,COL3
0187' 3C                      INC    A
0188' 67                COL3:   LD   H,A
0189' 22 ** **          LD    (FLSIZE),HL
018C'
018C' 3E 01                   LD     A,1
018E' ED 5B ** **             LD     DE,(DEBUF)
0192' 2A 64 1F          LD    HL,(_DTBUF)
0195' CD 00 20          CALL  _DRDSB
0198' D8                      RET    C
0199'
0199' 21 ** **          LD    HL,FILE_BF
019C' ED 5B ** **             LD     DE,(HLBUF)
01A0' 01 20 00          LD  . BC,20H
01A3' ED B0                   LDIR
01A5'
01A5' 3E 01                   LD     A,1
01A7' ED 5B ** **             LD     DE,(DEBUF)
01AB' 2A 64 1F          LD    HL,(_DTBUF)
01AE' CD 03 20          CALL  _DWTSB
01B1' D8                      RET    C
01B2'
01B2' CD ** **          CALL  RDFAT
01B5' D8                      RET    C
01B6'
01B6' 06 10                   LD     B,10H
01B8' 21 ** **          LD    HL,TBLCLST
01BB' 7E                COL1:   LD   A,(HL)
01BC' FE 7F                   CP     7FH
01BE' D2 ** **          JP    NC,WRFAT
01C1'
01C1' 23                      INC    HL
01C2' 4E                      LD     C,(HL)
01C3' E5                      PUSH   HL
01C4' 2A 62 1F          LD    HL,(_FATBF)
01C7' 16 00                   LD     D,0
01C9' 5F                      LD     E,A
01CA' 19                      ADD    HL,DE
01CB' 71                      LD     (HL),C
01CC' E1                      POP    HL
01CD'
01CD' 05                      DEC  · B
01CE' CA ** **          JP    Z,RDOPN3     ;Bad File Allocation
01D1' 18 E8                   JR     COL1
01D3'
01D3'
01D3'                   ;=================
01D3'                   ; INPUT  FROM FILE
01D3'                   ;=================
01D3' 2A ** **  INPUT_::LD    HL,(flvl)
01D6' 29                      ADD    HL,HL
01D7' 11 ** **          LD    DE,RDPNT
01DA' 19                      ADD    HL,DE        ;HL = RDPNT + (flvl) * 2
01DB' 34                      INC    (HL)
01DC' 20 1B                   JR     NZ,INP1      ;INC (RDPNT) lower byte
01DE'
01DE' E5                      PUSH   HL
01DF' 01 38 00          LD    BC,INF_SIZE
01E2' 2A ** **          LD    HL,(flvl)
01E5' CD ** **          CALL  MUL
01E8' 11 ** **          LD    DE,wkso
01EB' 19                      ADD    HL,DE        ;HL = wkso + INF_SIZE * (flvl
)
01EC' E5                      PUSH   HL
01ED' CD ** **          CALL  READ
01F0' E1                      POP    HL
01F1' 38 0D                   JR     C,INP2
01F3' 11 32 00          LD    DE,32H
01F6' 19                      ADD    HL,DE        ;HL points FLPNT
01F7' 34                      INC    (HL)
01F8' E1                      POP    HL
01F9'
01F9' 7E                INP1:   LD   A,(HL)
01FA' 23                      INC    HL
01FB' 66                      LD     H,(HL)
01FC' 6F                      LD     L,A
01FD' 7E                      LD     A,(HL)
01FE' B7                      OR     A            ;RCF
01FF' C9                      RET
0200'
0200' E1                INP2:   POP  HL
0201' C9                      RET                 ;RET with CY = 1 (Error)
0202'
0202' 00 B8             RDPNT:: DW   RDBUF
0204' 00 B9                   DW     RDBUF + 100H
0206' 00 BA                   DW     RDBUF + 200H
0208' 00 BB                   DW     RDBUF + 300H
020A'
020A' 01 38 00  READ:   LD    BC,INF_SIZE
020D' 22 ** **          LD    (FCB_ADR),HL
0210' 11 ** **          LD    DE,FILE_BF
0213' ED B0                   LDIR
0215'
0215' 3A ** **          LD    A,(FLDSK)
0218' 32 5D 1F          LD    (_DSK),A
021B'
021B' 3A ** **          LD    A,(FLPNT)
021E' 47                      LD     B,A
021F' 3A ** **          LD    A,(RC)
0222' B8                      CP     B
0223' DA ** **          JP    C,RDOPN3     ;Bad AllocationFile
0226'
0226' 78                      LD     A,B
0227' CD ** **          CALL  PNTREC
022A' EB                      EX     DE,HL        ; DE = RECORD
022B'
022B' 3A ** **          LD    A,(flvl)
022E' C6 B8                   ADD    A,0B8H       ;;RDBUF / 100H
0230' 67                      LD     H,A
0231' 2E 00                   LD     L,0
0233' 3E 01                   LD     A,1
0235' CD 00 20          CALL  _DRDSB
0238' D8                      RET    C
0239'
0239' 01 38 00          LD    BC,INF_SIZE
023C' ED 5B ** **             LD     DE,(FCB_ADR)
0240' 21 ** **          LD    HL,FILE_BF
0243' ED B0                   LDIR
0245' B7                      OR     A            ;CY = 0
0246' C9                      RET
0247'
0247'                   ;==============
0247'                   ; PRINT  TO FILE
0247'                   ;==============
0247'                   ;
0247'                   ;For REL-file
0247'                   ;
0247' 2A ** **  PRINTr::LD    HL,(wkre+12H)
024A' 23                      INC    HL
024B' 22 ** **          LD    (wkre+12H),HL      ;(FLSIZE)++
024E'
024E' 2A ** **          LD    HL,(WRPNT1)
0251' 77                      LD     (HL),A
0252'
0252' 21 ** **          LD    HL,WRPNT1
0255' 34                      INC    (HL)         ;INC (WRPNT1) lower byte
0256' 37                      SCF
0257' 3F                      CCF                 ;CY = 0
0258' C0                      RET    NZ
0259'
0259' 21 00 BC          LD    HL,WRBUF1
025C' 22 ** **          LD    (BUFAD),HL
025F' 21 ** **          LD    HL,wkre
0262' CD ** **          CALL  WRITE
0265' D8                      RET    C
0266'
0266' 21 ** **          LD    HL,wkre+32H  ;HL points FLPNT
0269' 34                      INC    (HL)
026A' C0                      RET    NZ           ;RET With CY = 0
026B' 36 FF                   LD     (HL),0FFH
026D' 37                      SCF
026E' 3E 05                   LD     A,5          ;Bad Record Error
0270' C9                      RET
0271'
0271' 00 BC             WRPNT1::DW   WRBUF1
0273'
0273'                   ;
0273'                   ;For PRN-file
0273'                   ;
0273' 2A ** **  PRINTp::LD    HL,(wkpr+12H)
0276' 23                      INC    HL
0277' 22 ** **          LD    (wkpr+12H),HL      ;(FLSIZE)++
027A'
027A' 2A ** **          LD    HL,(WRPNT2)
027D' 77                      LD     (HL),A
027E'
027E' 21 ** **          LD    HL,WRPNT2
0281' 34                      INC    (HL)
0282' 37                      SCF
0283' 3F                      CCF                 ;CY = 0
0284' C0                      RET    NZ
0285'
0285' 21 00 BD          LD    HL,WRBUF2
0288' 22 ** **          LD    (BUFAD),HL
028B' 21 ** **          LD    HL,wkpr
028E' CD ** **          CALL  WRITE
0291' D8                      RET    C
0292'
0292' 21 ** **          LD    HL,wkpr+32H  ;HL points FLPNT
0295' 34                      INC    (HL)
0296' C0                      RET    NZ           ;RET With CY = 0
0297' 36 FF                   LD     (HL),0FFH    ;FLPNT = 0FFH
0299' 37                      SCF
029A' 3E 05                   LD     A,5          ;Bad Record Number
029C' C9                      RET
029D'
029D' 00 BD             WRPNT2::DW   WRBUF2
029F'
029F'
029F'
029F' 01 38 00  WRITE:  LD    BC,INF_SIZE
02A2' 11 ** **          LD    DE,FILE_BF
02A5' 22 ** **          LD    (FCB_ADR),HL
02A8' ED B0                   LDIR
02AA'
02AA' 3A ** **          LD    A,(FLDSK)
02AD' 32 5D 1F          LD    (_DSK),A
02B0'
02B0' 3A ** **  WRITE1: LD    A,(FLPNT)
02B3' 47                      LD     B,A
02B4' 3A ** **          LD    A,(RC)
02B7' B8                      CP     B
02B8' 30 76                   JR     NC,WRITE4    ;if (FLPNT) <= ((RC) ,JP WRIT
E4
02BA'
02BA' CD ** **          CALL  RDFAT
02BD' D8                      RET    C
02BE'
02BE' 3A ** **          LD    A,(FLPNT)
02C1' E6 F0                   AND    0F0H
02C3' 47                      LD     B,A
02C4' 3A ** **          LD    A,(RC)
02C7' E6 F0                   AND    0F0H
02C9' B8                      CP     B
02CA' 30 47                   JR     NC,WRITE2
02CC'
02CC' CB 3F                   SRL    A
02CE' CB 3F                   SRL    A
02D0' CB 3F                   SRL    A
02D2' CB 3F                   SRL    A            ;A = A / 16
02D4' 21 ** **          LD    HL,TBLCLST+1
02D7' 16 00                   LD     D,0
02D9' 5F                      LD     E,A
02DA' 19                      ADD    HL,DE
02DB' E5                      PUSH   HL
02DC'
02DC' 3A ** **          LD    A,(RC)
02DF' E6 F0                   AND    0F0H
02E1' CD ** **          CALL  PNTREC
02E4' CD ** **          CALL  RECCL
02E7'
02E7' 2A 62 1F          LD    HL,(_FATBF)
02EA' 16 00                   LD     D,0
02EC' 5F                      LD     E,A          ;A = RC && 0F0H
02ED' 19                      ADD    HL,DE
02EE' 36 8F                   LD     (HL),8FH     ;DUMMY
02F0' CD ** **          CALL  FCGET
02F3' 77                      LD     (HL),A
02F4' E1                      POP    HL
02F5' D8                      RET    C
02F6' 77                      LD     (HL),A
02F7' 23                      INC    HL
02F8' 36 80                   LD     (HL),80H
02FA'
02FA' 2A 62 1F          LD    HL,(_FATBF)
02FD' 16 00                   LD     D,0
02FF' 5F                      LD     E,A
0300' 19                      ADD    HL,DE
0301' 36 80                   LD     (HL),80H
0303' 3A ** **          LD    A,(RC)
0306' E6 F0                   AND    0F0H
0308' C6 10                   ADD    A,16         ;ONE CLUSTER ADDED
030A' 32 ** **          LD    (RC),A
030D'
030D' CD ** **          CALL  WRFAT
0310' D8                      RET    C
0311' 18 9D                   JR     WRITE1
0313'
0313' 3A ** **  WRITE2: LD    A,(FLPNT)
0316' 32 ** **          LD    (RC),A
0319' CB 3F                   SRL    A
031B' CB 3F                   SRL    A
031D' CB 3F                   SRL    A
031F' CB 3F                   SRL    A            ;A = A / 16
0321' 21 ** **          LD    HL,TBLCLST+1
0324' 16 00                   LD     D,0
0326' 5F                      LD     E,A
0327' 19                      ADD    HL,DE
0328'
0328' 3A ** **.         LD    A,(FLPNT)
032B' E6 0F                   AND    0FH
032D' C6 80                   ADD    A,80H
032F' 77                      LD     (HL),A
0330'
0330' 3A ** **  WRITE4: LD    A,(FLPNT)
0333' CD ** **          CALL  PNTREC
0336' EB                      EX     DE,HL        ;DE = RECORD NO.
0337'
0337' 3E 01                   LD     A,1
0339' 2A ** **          LD    HL,(BUFAD)
033C' CD 03 20          CALL  _DWTSB
033F' D8                      RET    C
0340'
0340' 01 38 00          LD    BC,INF_SIZE
0343' ED 5B ** **             LD     DE,(FCB_ADR)
0347' 21 ** **          LD    HL,FILE_BF
034A' ED B0                   LDIR
034C' B7                      OR     A            ;RCF
034D' C9                      RET
034E'
034E'                   BUFAD:: DS   2
0350'
0350'
0350'                   ; FILE POINTER (A) => RECORD NO. (HL)
0350' F5                PNTREC: PUSH AF
0351' F5                      PUSH   AF
0352' CB 3F                   SRL    A
0354' CB 3F                   SRL    A
0356' CB 3F                   SRL    A
0358' CB 3F                   SRL    A            ;A = A / 16
035A' 21 ** **          LD    HL,TBLCLST
035D' 16 00                   LD     D,0
035F' 5F                      LD     E,A
0360' 19                      ADD    HL,DE
0361' 7E                      LD     A,(HL)
0362' CD ** **          CALL  CLREC
0365' F1                      POP    AF
0366' E6 0F                   AND    0FH
0368' 85                      ADD    A,L
0369' 6F                      LD     L,A
036A' F1                      POP    AF
036B' C9                      RET
036C'
```

```
036C'
036C'                  ; FAT READ TO BUFFER
036C' D5               RDFAT:  PUSH   DE
036D' E5                       PUSH   HL
036E' 3A ** **         LD      A,(LST_DSK)
0371' 57                       LD     D,A
0372' 3A 5D 1F         LD      A,(_DSK)
0375' BA                       CP     D
0376' 28 0F                    JR     Z,RDFAT_1
0378' 32 ** **         LD      (LST_DSK),A
037B' 3E 01                    LD     A,1
037D' ED 5B 5E 1F              LD     DE,(_FATPOS)
0381' 2A 62 1F         LD      HL,(_FATBF)
0384' CD 00 20         CALL    _DRDSB
0387' E1               RDFAT_1:POP    HL
0388' D1                       POP    DE
0389' C9                       RET
038A'
038A'                  ; FAT WRITE FROM BUFFER
038A' D5               WRFAT:  PUSH   DE
038B' E5                       PUSH   HL
038C' 3E 01                    LD     A,1
038E' ED 5B 5E 1F              LD     DE,(_FATPOS)
0392' 2A 62 1F         LD      HL,(_FATBF)
0395' CD 03 20         CALL    _DWTSB
0398' E1                       POP    HL
0399' D1                       POP    DE
039A' C9                       RET
039B'
039B'                  ; FREE CLUSTER POSITION    GET
039B' C5               FCGET:  PUSH   BC
039C' E5                       PUSH   HL
039D' 06 80                    LD     B,80H
039F' 2A 62 1F         LD      HL,(_FATBF)
```

```
03A2' 7E               FCGET2: LD     A,(HL)
03A3' B7                       OR     A
03A4' 28 08                    JR     Z,FCGET3
03A6' 23                       INC    HL
03A7' 10 F9                    DJNZ   FCGET2
03A9' 3E 09                    LD     A,9          ;DeviceFull
03AB' 37                       SCF
03AC' 18 04                    JR     FCGET4
03AE' 3E 80            FCGET3: LD     A,80H
03B0' 90                       SUB    B
03B1' A7                       AND    A            ;CY = 0
03B2' E1               FCGET4: POP    HL
03B3' C1                       POP    BC
03B4' C9                       RET
03B5'
03B5'                  ; CLUSTER (A) => RECORD     (HL)
03B5' 26 00            CLREC:  LD     H,0
03B7' 6F                       LD     L,A
03B8' 29                       ADD    HL,HL
03B9' 29                       ADD    HL,HL
03BA' 29                       ADD    HL,HL
03BB' 29                       ADD    HL,HL
03BC' C9                       RET
03BD'
03BD'                  ; RECORD (HL) => CLUSTER (A)
03BD' E5               RECCL:  PUSH   HL
03BE' CB 3C                    SRL    H
03C0' CB 1D                    RR     L      ;HL/2
03C2' CB 3C                    SRL    H
03C4' CB 1D                    RR     L      ;HL/4
03C6' CB 3C                    SRL    H
03C8' CB 1D                    RR     L      ;HL/8
03CA' CB 3C                    SRL    H
03CC' CB 1D                    RR     L      ;HL/16
```

```
03CE' 7D                       LD     A,L
03CF' E1                       POP    HL
03D0' C9                       RET
03D1'
03D1'
03D1'                  FCB_ADR:DS     2
03D3'                  LST_DSK:DS     1
03D4'
03D4'                  ;
03D4'                  ; Works
03D4'                  ;
03D4'                          DSEG
0000"
0038                   INF_SIZE EQU   38H
0000"
0000"                  FILE_BF:DS     12H
0012"                  FLSIZE:  DS    2   ;File Size.
0014"                  FLDTADR:DS     2   ;Start Address.
0016"                  FLEXADR:DS     2   ;Exec Address.
0018"                          DS     6
001E"                  FSTCLST:DS     1   ;First Cluster.
001F"                          DS     1
0020"                  TBLCLST:DS     11H.;Cluster table.
0031"                  FLDSK:   DS    1   ;The Login disk.
0032"                  FLPNT:   DS    1   ;The FILE Pointor.
0033"                  DEBUF:   DS    2   ;Record No.Which Have The DIR.
0035"                  HLBUF:   DS    2   ;Address Where On INFORMATION.
0037"                  RC:      DS    1   ;The Numberof Records The File have.
0038"
0038"                  WKend::             ;The Mark of the Ending address of Wor
k Area
0038"                          END
```

全機種共通システムインデックス

■85年6月号
序論　共通化の試み
第1部　S-OS"MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶ magiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版 S-OS"SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版 S-OS"SWORD"
■86年9月号
第28部　FuzzyBASIC 発表
連載　明日に向かって magiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタ DREAM
第31部　FuzzyBASIC 料理法〈1〉
■86年11月号
第32部　パズルゲーム HOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC 料理法〈2〉
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC 料理法〈3〉
■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC 料理法〈4〉
■87年2月号
第36部　アドベンチャーゲーム MARMALADE
第37部　テキアベ作成ツール CONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツール MAGE
付録　"SWORD" 再掲載と MAGIC の標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS"SWORD" 変身セット
第43部　MZ-700用 "SWORD" を QD 対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASIC コンパイラ
第45部　エディタアセンブラ ZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージ JACKWRITE
特別付録　FM-7/77版 S-OS"SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラ Inside-R
特別付録　PC-8001/8801 版 S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASIC コンパイラの拡張
第52部　X1turbo 版 S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OS の仲間たち
第53部　もうひとつの FuzzyBASIC 入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OS こちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo 版 "SWORD" アフターケア ラインプリントルーチン
特別付録　PASOPIA7 版 S-OS"SWORD"
■88年1月号
第58部　FuzzyBASIC コンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲーム ELFES
■88年3月号
第60部　構造型コンパイラ言語 SLANG
■88年4月号
第61部　デバッギングツール TRADE
第62部　シミュレーションウォーゲーム WALRUS
■88年5月号
第63部　シューティングゲーム ELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語 SLANG 入門(1)
第66部　Lisp-85 用 NAMPA シミュレーション
■88年7月号
第67部　マルチウィンドウドライバ MW-1
連載　構造化言語 SLANG 入門(2)
■88年8月号
第68部　マルチウィンドウエディタ WINER
■88年9月号
第69部　超小型エディタ TED-750
第70部　アフターケア WINER の拡張
■88年10月号
第71部　SLANG 用ファイル入出力ライブラリ
第72部　シューティングゲーム MANKAI
■88年11月号
第73部　シューティングゲーム ELFES Ⅳ
■88年12月号
第74部　ソースジェネレータ SOURCERY
■89年1月号
第75部　パズルゲーム LAST ONE
第76部　ブロックゲーム FLICK
■89年2月号
第77部　高速エディタアセンブラ REDA
特別付録　X1版 S-OS"SWORD"〈再掲載〉
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOROBAN
■89年4月号
第79部　SLANG 用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータ RING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲーム　TICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲーム PUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリ DIO. LIB
■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲーム SQUASH！
第95部　X68000対応S-OS"SWORD"
特別付録　PC-286対応S-OS"SWORD"

＊以上のアプリケーションは，基本システムであるS-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

復活のCGアニメーション

プロジェクトチーム DōGA かまた ゆたか

今回は，そろそろ本格的な作品制作に取り掛かろうという方のために，CGA作品の企画について考えてみたいと思います。また，めでたい連載1周年を記念して，バージョンアップサービスのお知らせも掲載いたします。

はじめに

2カ月連続して，連載を休んで申し訳ありません。“あまりの人気のなさで連載が打ち切られた？”“かまたが過労（あるいは急性胃潰瘍）で倒れた？”などいろいろ噂も飛び交い，当方にも何件かの問い合わせがありました。しかし実際には，特にこれといった理由はなく，文字どおり“申し訳がナイ”のです。今月はちょうど1周年ということもありますので，一部の連載陣も交代して，気分一新頑張っていきたいと思います。

さて，5月号に「アマチュアCGAコンテスト入選作品集」ビデオテープ配布のお知らせを掲載しましたが，皆さん申し込まれたでしょうか（すでに締め切りましたので，見落としていた方はご了承ください）。このビデオをご覧になれば，現時点でのアマチュアCGAのレベルというものがはっきりわかると思います。皆さんは，“この程度なら私のほうが上だ”と思いましたか？　それとも“オォ！　私ニハ勝テナイ（出典：アジオージャ）”と諦めてしまいましたか？　とりあえずCGAシステムで適当にカットを作れるようになってきた方でも，コンテストに応募するような“作品”という形にはまとめられないとか，ストーリーを思いつかないという方は多いのではないでしょうか。

今年度のCGAコンテストの締め切りも，12月31日です。夏休みに本格的な作品制作を行うために，そろそろ作品構想に入らなければいけません。ということで，今回はCGAの作品企画を取り上げます。

作品企画

作品企画のメインは，もちろん制作する作品の内容を決めることです。そのほかに，制作のスケジュールも決めなければいけません。また，チームなどで共同制作する場合は各自の分担を決め，効率よく制作できるように準備します。作品の内容の決め方は，作品の種類や目的によってまったく異なり，画一的な手法があるわけではありませんので，のちほどいくつかのパターンをまとめて紹介します。

それではまず，CGA作品企画において最も大切なポイントについてお話ししましょう。それはズバリ“作品を完成させる”ことです。これを常に最優先にすることを念頭においてください。作品が完成しなければ，それまで行ったすべての努力とアイデアが水の泡と化してしまうのです。

“作品を完成させる”企画を立てるためには，自分の持っているパワーを正しく認識する必要があります。自分の持っているパワーというのは，単に“努力と根性”ということではなく，技術的なパワー，CGA制作に割ける時間的問題，自分の持っているハード上の制約，制作に参加する人数と各自のパワーなども含みます。誰しも作品企画の段階では，いままでになかったような凄い作品を作ってやろうと思うものです。そういった意気込みは大変大切なもので，そのくらいでなければまともな作品にはならないのです。が，凄い作品には凄いパワーが必要になるので，結局はオーバーワークになり，“未完の大作”となってしまいがちです。今回制作するのは，凄い作品を作るための習作なのだと自分にいい聞かせて，“ちょっと努力すれば，すぐできてしまう”と思える程度の作品を企画するのがコツでしょう。

作品を完成させるためにもうひとつ考えなければいけないのは，目的をはっきりさせることです。学園祭で上映するための作品なのか，新入生募集のPR用なのか。別に期限などない気ままな作品というのもあるでしょうが，期限のない作品は完成した試しがありません。いつまでに必ず完成しなければいけないという必然性を持つことは非常に大切です。最初はすばらしいと思った企画でも，制作の過程では，なかなか作業がはかどらなかったり，思ったようなイメージが表現できなかったりして，だんだん嫌になってくる時期がきます。そんな時期を乗り越えて完成させてこそ，作品制作の喜びが体験できるわけですから，途中でくじけないために，この必然性を企画の段階から考えてください。

特に目的がない方は，当方が主催する「アマチュアCGAコンテスト」での入選を目標に励んでください。そうすれば，期限は自動的に今年の年末になります。そのほか，友達の前で，“何月何日までにこの作品が完成しなければ，私のX68000をDōGAにカンパしてやる”と宣言し，自らを窮地に追い込むのもよいでしょう。でもこの場合，当方としては完成しないほうがうれしく思います。

次にスケジュールの組み方ですが，これは目的（期限）と自分のパワーがわかればだいたい決まってしまいます。要は，スケジュールなんて所詮遅れるものなのだから，真剣に期限までに完成させる気があるのなら，余裕を持

って計画すればよいのです。以下に，サンプルを挙げますので参考にしてください。

条件1：CGAコンテストに出品する
条件2：7月後半から9月初頭は夏休みで余裕がある
条件3：9月後半から10月初頭は試験で時間がない

企画（絵コンテ含む）	6月下旬～7月下旬
モデリング	7月上旬～7月下旬
モーションデザイン	7月下旬～8月下旬
レンダリング	8月上旬～9月中旬
編集/仕上げ	10月中旬～11月初頭
試写会	11月初頭
改良	11月上旬～11月下旬

企画からレンダリングまでの作業は，お互いにオーバーラップしてますが，年末に完成するのなら，8月に入る前にすべての形状デザインを，9月に入る前にモーションデザインを終えておくべきだと解釈してください。この例では，余裕を持たせて，コンテストの締切のひと月前に完成することになっていますが，その程度の余裕では，締切の前夜は徹夜することを覚悟したほうがよいでしょう。11月初頭の試写会というのは，大阪大学の学祭のことですが，コンテストで本気に入賞を狙うのなら，ひと月前には最初のバージョンを完成させておいて，ほかの人の意見を聞いてみることをお勧めします。試写会の場がなければ，当チームにお送りくださいますと，寺田からの“教育的指導”があるでしょう。

作品企画例

1）かおる流　明日は明日の風が吹く法

数年前に，かおる君が制作した「ずんぐり＆むっくり」は，エンタープライズもどきの“ずんぐり号”とリライアントもどきの“むっくり号”との戦いをコミカルに，メルヘンチックに描いた心温まる(?)作品です。宇宙空間をすっとばしていたずんぐり号の前に，むっくり号が現れ，びっくりしたずんぐり号は急停止します。続いて，むっくり号はフェイザーを発射してきたので，ずんぐり号も光子魚雷をお見舞いしてやりました。光子魚雷を受けたむっくり号は，小さな3つのむっくり号に分裂してしまいます。そこにずんぐり号が近寄って，放り投げたり，背中に乗せたりして遊びます。再び3つのむっくり号は合体して，今度はずんぐり号と同じ姿になってしまい，2台仲よく宇宙の彼方へ飛んでいきます。

この想像を絶するわけのわからないストーリー展開はどのようにして生まれたのでしょうか。実は，この作品においては企画なんてものは存在しません。とりあえず，1カットを作り，それができたあとで，次の1カットをどのようにするかを考えるという，実にいいかげんな方法で制作されています。初心者なのに，何月何日までにとりあえず何か見せるものを作らないといけないという方にはよい方法といえます。また，どうしても作品を思いつかないときの最後の手段としても有効でしょう。コツは，とにかく次の1カットのことしか考えない。次の1カットをいかに面白く，あるいは意外性をもたせるかだけを考えて，どんどんつなげていくのです。そして，ある程度の量ができたら，あるいは締切が近づいてきたところで，強引にオチをつけて，終わりにしてしまえばよいのです。この方法のよいところは，誰でもすぐに作品制作に掛かれるということと，一応ひとつの作品を完成することができるという点です。ただし，かなり偶然性に頼っているわけで，傑作ができる可能性はあまり多くないかもしれません（作者のセンス次第ですが）。

2）げんし流　ぱちもん法

「Mの喜劇」という，げんし君が制作したこれまた古い作品があります。これは，アカデミー短編賞にもノミネートされたかの名作，PIXAR社の「Luxo Jr.」のパロディです。非常に有名な作品なのでご存じの方も多いと思いますが，原作では電気スタンドの親子とゴムまりが出てくるところを，「Mの喜劇」ではパソコンの親子とピラミッドに変えています。やはり，げんし君が「Luxo Jr.」の大ファンというのが制作の動機のようです。

また，チームTOSAKAがCGAシステムで制作した「スタートレック・カーンの圧勝（CGAコンテストのオープニング）」は，映画版スタートレックのパロディです。さらに，作者は存じませんがガンダムのキャラクターがSTAR WARSのデススターに突入するという「ハロZZ」という作品も見たことがあります。

これらの作品は，自分にとって特別思い入れのある既存の作品を自分の手でCG化してしまうというパターンで，異様なパワーを発揮し，技術的にも完成度も高いという共通点があるようです。その異様なパワーの源がなんにしろ，自分の作品に対してのめり込めるというのは大変よいことで，この手の作品の大きな強みです。

ただ，パロディである分だけオリジナリティは少なく，

初めて読む読者へ

この連載では，DōGA・CGAシステムを中心にして，CGアニメーション作品の作り方を取り上げています。DōGAとは何か？　CGAシステムとは何か？　簡単におさらいしておきましょう。

プロジェクトチームDōGAとは，“手軽でパーソナルな映像としてのCGアニメーションの普及”を目的に，大阪大学コンピュータクラブや京都大学マイコンクラブが中心になって活動しているアマチュアの団体です。そしてCGAシステムとは，DōGAにて開発された，パソコン（X68000）上で手軽にCGA作品制作を行うためのプログラムです。このCGAシステムを，私たちの活動に賛同するアマチュアに限って，実費＋カンパで配布すると本誌上でも発表したところ全国から異様にたくさんの申し込みをいただき，アマチュアCGAの活動は一気に全国に広がりました。そこで，当チームの活動もシステム開発だけでなく，CGA作品制作，上映会，アマチュアCGAコンテストの開催，全国のアマチュアCGA団体との連携など幅広い活動を展開しているのです。

なお，本連載の経過は以下のとおりです。入手ご希望の方は，本誌バックナンバー案内をご覧ください。

コンテストを目的とした場合は多少不利になるのはしかたないでしょう。しかし，逆にいえば新歓のPRなどが目的の作品ならば，全然気にする必要がないということです。また，パロディでもオチを変えたりしてオリジナリティを出すということは十分可能だと思います。

この手の作品を制作すると，原作に近づけようと努力する過程で，一流作品の“見せ方”や“編集”のテクニックを知らず知らずのうちに身につけるという，大きなメリットがあります。自分の手でCG化してみたい映像に心当たりのある方は，一度トライしてみるのもよい勉強となるでしょう。

3） 雪だるま流　BGM法

「冬の終わる夜」は，全日本ビデオコンテストへの出品と，女性へのCGAのPRという打算に満ちた意図で制作されました。まず最初に夢で見たイメージがあり，それを音楽担当者に伝えて作曲してもらいました。具体的な内容はすべてこの音楽に合わせて考えていきます。

この，BGMを先に決定してそれに合わせて映像を企画するという手法は，非常に汎用性があり有効です。音楽によってイメージがはっきりしているので，ストーリーや映像を想像しやすく，自然と全体がまとまります。極端な話，脈絡のないカットのつなぎ合わせのようにストーリー性のないものでも，音楽に合っていれば，ひとつの作品になるといってもウソではありません。センスがよければ，観客に感動を与える名作にもなり得ます。さらに，通常のメディアと違ってCGAは，コンマ１秒単位の制御をデジタルに処理できるため，音楽にぴったり合わせるという作業も比較的楽です。

問題点としては，まず著作権問題が挙げられます。個人として楽しむならいざ知らず，コンテストに出品する場合主催者側に曲の変更を求められる場合もあり，作品の魅力が半減します。アマチュアCGAコンテストでも，著作権問題がクリアされていない作品は基本的にお断りです。このテの作品の場合，音楽は最も重要な要素になるのですから，オリジナル曲で頑張ってもらいたいと思います。しかし，オリジナル曲ができないのであれば，BGM用の著作権が放棄された曲（レンタルレコード店にあるそうです）を利用するとか，ちゃんと著作権協会に許可を取るようにしましょう。もうひとつの問題は，作品の長さなどが強制的に曲に縛られるという点です。音楽をMIDIデータとして持ち演奏させるなら，映像の都合に合わせて一部を編集することも可能となります。また，曲をフェードアウトで強引に終わらせてもよいでしょう。

この手法の場合，当然企画は音楽を決めるところから始まります。それはもう，かたっぱしから音楽を聴くしかありません。自分の好き嫌いよりも，映像を思い浮か

各読者通達事項

バージョンアップサービスのお知らせ

連載１周年を記念して，CGAシステムのバージョンアップサービスを行います。新しいバージョンは「2.11」あるいは「2.12」です。「2.0*」との主な違いは下記のとおりです。あまり根本的な変更はないのですが，なかには役に立つ機能やプログラムもありますので，ご希望の方は申し込んでください。

今回のサービスの内容は，マニュアルなどもなく，ディスクに変更のあったプログラムだけを入れて送るという簡単なものですので，実費は500円程度ですむと思われます。前々からの約束どおり，バージョン「2.00」および「2.01」をお持ちの方には基本的に無料（カンパはご自由に）とさせていただきます。

なお，サービスのスケジュールは以下のようになる予定です。

申し込み開始	６月18日
申し込み締切	７月31日
発送開始	８月下旬
実費等の払い込み	受け取り後２週間以内（用紙は同封します）
苦情（未着等）対応	９月中旬～９月30日

●バージョンアップの申し込み方法

必ず官製ハガキに，次の点を明記して当プロジェクトルームまでお送りください。記入ミスなどによるトラブルには対応致しかねますし，“マニュアルを送れ”“２セット送れ”“コンテストのビデオまだありますか？”といった，別の用件をいっしょに書いてこられても，個別対応はできませんのでご了承ください。

１）氏名
２）住所（郵便番号）
３）電話番号
４）お手持ちのCGAシステムのバージョン
５）下記のアンケート（複数回答可）

アンケート

このCGアニメーション講座の連載において，どのような内容を望みますか？

a）マニュアルの代わりとなるようなCGAシステムの使い方
b）マニュアルを読んでもわからない人のための入門講座
c）マニュアルに載っていないような高度なテクニック
d）CGA作品制作法
e）DōGAにおける作品制作の実況レポート
f）映像制作一般論
g）CGのアルゴリズムやプログラム開発方法
h）イベントレポートなどのDōGAの活動報告
i）その他（ご自由に記入してください）

●主なバージョンアップの内容

・REND，FFE，ATRの各プログラムがバージョンアップした。

REND	：バックグラウンド機能，若干の高速化，背景呼び込み，空気遠近法
FFE	：ロード機能，若干の高速化
ATR	：材質感表示が可能，操作性を一新

・Ver2.0* を発表してから新しく作られたプログラム（HANIM，IC，CRD，SMOOTH，SUFCUT，STAR，BOMB，PATIPIC）が入った。

HANIM	：SRANIMを２倍高速化（ただし65536色中256色）
IC	：512の画像データの一部（256）を連続的に切り出す
CRD	：65536色の画像データを256色に落とす
SMOOTH	：テロップ画像データのアンチエイリアス化
SUFCUT	：形状データの一部を切り出し，部品化する
STAR	：背景用の星空を生成する
BOMB	：形状データを爆発させる
PATIPIC	：Z'sSTAFFに常駐し，CGAシステム用の画像ファイルを出力する

・CADが新しいIOCS（SX-WINDOWについているもの）に対応した。

・Humanのシステムが Ver2.01になった。

なお，Ver2.10とはいくつかのバグがとれた程度で，基本的には変っていません。

X68000最新機種をお持ちの方へ

PROII，EXPERTII，SUPER-HDなどの最新機種では，IOCSが変更されており，CGAシステムの一部のプログラムを使用したとき不都合が出ることがわかりました。また，旧機種に最新の「IOCS.X」を組み込んだ場合も同様です。とはいっても，マウスカーソルのあとが残るときがあるという程度のもので，まったく使えなくなるわけではありませんし，若干高速化されるので，私は気にせずに使っています。

今回のバージョンアップで対応しようと思っていたのですが，問題のあるすべてのプログラムを全部直すのは間に合わないと思います。一応の対応策を記しますので，ご注意ください。

・気にしない
・CGAシステムディスクから起動する
・CONFIG.SYSの中でIOCS.Xを組み込むのをやめる

べやすいかどうかという基準に重点をおきます。また，イメージがひんぱんに変わってくるような曲も，映像をぴったり合わせるのが難しいのでやめておいたほうがよいでしょう。それから，歌詞にはあまり捕らわれないでよいと思います。外国の歌詞なら，どうせ誰もわからないし，日本語の歌詞でどうしてもじゃまになるようでしたら，カラオケを利用するのもひとつの手です。チームで制作するときは，ほかのスタッフに話す前に曲を決めておくほうがよいでしょう。曲によってイメージを共有できるのはこの手法の利点といえます。

曲を決めたら，何度も何度も聴き，映像のイメージを固めていきます。最終的には，その曲を聴けば頭の中で全カットの映像が浮かぶまで詰めます。そして，楽譜を見ながら，“この音が入る瞬間にカメラはここを向いているから……”という具合に絵コンテを制作すると，フレーム数まではっきりするコンテができるでしょう。

4） 令子流　4カット法

これは前々から一度チャレンジしようと思っていたトッテオキの企画ですが，4コマ漫画と同じように，起・承・転・結をそれぞれ1カットで表現し，4カットのみでひとつの作品を構成する手法です。ですから1カットが5秒程度とすると，タイトルを入れても30秒足らずの小作品ということになります。短いだけあって，制作は簡単ですし，起承転結の構成を身につける練習にもなります。いくらなんでも30秒というのは短すぎるというのでしたら，同じ主人公キャラを使用した作品を毎月1作ずつ制作し，年末には6本まとめたオムニバスという形式を取ればよいのです。そうすると，途中で急用ができ，制作が1，2カ月遅れても，本数は少なくなってもちゃんと作品は完成するというメリットもあります。それに，新入部員に対する練習課題にも適しています。

問題点は当チームでも試してみたことがないからわかりません。やはりオチが決まらないと見るに耐えないので，アイデアが勝負でしょう。どうしても，ギャグ，コメディが中心になると思いますが，まったく新しいジャンルを開拓する余地も多いと思います。

5） VOYAGER流　点と線法

昨年制作した「Thank you VOYAGER」では，海王星の影でVOYAGERが奮闘する様子を描くことで宇宙開発に携わっている方々への尊敬と感謝の念を表現しようと試みました。まず最初にイメージしたのが，“ゆっくりと回転する海王星に吸い込まれていくVOYAGER”のカットと，“VOYAGERから送られてくる分析図に宇宙開拓史がだぶってスクロールする”といったカットです。次に“太陽系を離脱し，暗い宇宙空間に消えていく”“BGMがフェードアウトし，無音で暗黒の影の部分に入っていく”“逆光を浴びながら影から出てくる”カットなどを考えました。これらは，すべて連続性のないバラバラのカットであり，ストーリーとしてつながっているわけではありません。すなわちこれが“点”つまり“見せ場”なのです。そして，“点”をたくさん考え，“線”でつなぐことでストーリーを作るのがこの“点と線”法です。

もう少し詳しく解説しましょう。まず，“点”を思いついたら，どんどん紙に描いていきます。必ず，ひとつの“点”につき，1枚の紙を用い，非常にラフなイラストと解説を添えます。この段階では，ストーリーとか，構成とかいうものを考える必要はなく，関連ありそうなカットやアイデアを蓄えます。約5分の「VOYAGER」で，1週間以上をこの作業に費やしました。次にそれらの紙を床にばらまき，スタッフとともにいろいろ並べてみましょう。“これとこれは一連のシーンとしてつながるのでまとめておこう”とか，“このカットは全然別なのでこっちの隅に置いておこう”とか，“これはエンディングに使えるから右端だ”とか分類できてきます。そしてその作業が進むと，だんだんいく通りかのストーリーが見えてくるでしょう。“このへんのとこのへんのを使って，このエンディングにもっていけば，3分ぐらいの短編ができそうだ”とか，“その順番を入れ替えて，その前にこの辺のをくっつけると話が膨らんで5分ぐらいの作品になる”という具合です。基本的には，できるだけ多くの

モデラー高津のLOGIN

突然ですが，松井さんが忙しくなったため，SIGOPを引退することになりました。かわりに5月からはこの私，モデラー高津がSIGOPをしています（といっても私が暇というわけではケッシテありません）。

ということで，その副作用により，今回からこのコーナーも“モデラー高津のLOGIN”となることになりました。私は知る人ぞ知るあの“パロレイバー”の作者です。DōGAにおいても，あるときはプログラミング，あるときはモデリング，月に2回「花とゆめ」を持ち込むなんてこともしています。苦情係も兼任しているので，文句のある方はどうぞ。私は誰の挑戦も受けます。

今後のネットでの予定ですが，従来どおり，ユーザーの質問に対するQ&A，バグレポート，新しいツールのアップはもちろん行っていきます。さらに，ユーザー同士の交流の場として活用してもらいたいと思っています。最近は，名古屋のチーム“鮮色映人”からアルファベットの形状データがアップされたり，ネットもなかなか活発です。このようなユーザーからのアプローチは，SIGの本来の目的であり，たいへん喜ばしいことですので，データのアップもどんどん増えてほしいものです。

ところで，最近TVアニメのロボットの形状データを作ったというお便りをよくいただきますので，著作権問題について専門家のご意見を伺ってみました。

DōGA「もしもし，日本サンライズですか？少々お伺いしますが，御社のTVアニメの形状データをネットにアップすると罰せられるでしょうか？」

日本サンライズ「厳密にいえば問題がないわけではありませんが，そのデータを有料で販売するなどしなければ，別にかまいません」

……ということですので，みなさんかっこいいデータを作ってアップしてください。

今月のアップデータ

今月は半端なものではありません。チームTOSAKA制作，CGAコンテストオープニングアニメーションに使用された「エンタープライズ&リライアント」です。これを語るのには「努力と根性」という言葉しかありません。ネットにアップするときは，OPアニメーションのワンカットのフレームソースと一緒にして載せるつもりなので，楽しみにしてください。

＊

CGAシステムに関して質問がございましたらDōGAプロジェクトルームの「当てにならないアフターサービス係」宛か，次の各ネットにメールで送ってください。それでは，またDōGA・CGA NETでお会いしましょう。

ハンドルネーム

Taka2	J&PHOTLINE	JH082274
	アスキーネット	PCSPCS33268
	PC-VAN	VAM28517

“点”を通るような“線”を作ればよいのです。が，無理に全部“線”上に乗せてねじ曲がったストーリーになっては意味がないので，せっかくのアイデアとはいえ，次回に利用しようと潔く諦めてください。

大筋が見えてきた時点でこの作業は終了し，もう一度“点”を作る作業に戻ります。なぜなら，このままでは“点”が密集した部分と，直接つながらないのを“線”で強引につなげただけのところの差が激しいからです。本当に“線”でつなげられるのかをもう一度チェックするとともに，その線上にくる“点”を見つけていかなければいけません。つなぎのカットばっかりが連続するシーンなんて，中だるみそのものです。

この“点と線”法は，ある意味で作品企画としていちばんオーソドックスな手法といえます。また，練習すればすべての作業を頭の中でできるようになるでしょう。

実録！ 作品企画

具体的に作品企画はどのように決定されるのか。4月21，22日にDōGAにて行われた作品企画会議のようすを紹介しますので，参考にしてください。

21日は，大阪大学コンピュータクラブの入部説明会があり，特にCGに興味のある新入部員はその日からいきなりDōGAのプロジェクトルームに連れてこられる。今回の企画会議は，この新入生にゴールデンウィークに制作してもらう作品を決めるもので，期間が短いこともあり“簡単にできる”という条件が初めから与えられていた。

まず最初に，作品制作のスタッフを決めなければいけない。“いきなり作品制作にチャレンジする者は誰かいないか”という問いに，いち早く手を挙げた3人組（以後，新米A，B，Cと呼ぶ）がスタッフに決定した。また，中学時代から作曲，演奏などをやっており，音楽だけではなくプロモーションビデオなどの映像も手掛けてみたいとやってきた新人M（Mは“MIDI”の“M”）に，一応の監督をしてもらうことにした。さらにこの4人の新入生をサポートする上級生として，寺田，MAX田口，モデラー高津，本連載では出番のなかった砂川が参加し，私もプロデューサとしてまとめ役をすることになった。

21日，新入生には，当チームで制作された作品やアマチュアCGAコンテストの作品を見てもらい，アマチュアCGAの現状やレベルを把握してもらった。また，CGAシステムの実演を交えながら，CGA作品制作の流れと，CGAシステムには表現しやすいものとそうではないものがあることを解説した。とりあえず新米3人組には，明日までに作品のアイデアを考えてくるようにいって，解散し，Mにいろいろ話を聞いてみることにした。監督とプロデューサとの意思疎通は大切なのだ。

Mが目指している映像は，音楽と同じように，テーマなどというものはなくても，かっこいいとか，面白いといった，感覚的なフィーリングやイメージで作るべきだという。私はそれには反対で，映像がコミュニケーションの手段である以上，制作者の意図やテーマというのは作品制作において重要な意味を持つと思う。

このように，作品企画の段階でスタッフ内の意見が食い違うことはよくある。また制作の過程においても，このカットは省略するかどうか，色を変えてレンダリングし直すかどうかなど，もめごとはいくらでも起こり得る。そんなときは，迷わず監督（あるいはリーダー）に一任しよう。そして，どうしても監督の指示に納得がいかない場合でも，とりあえずその作品を完成させるために協力する。そしてそのあとで，今度は自分が監督になって作品を作ればよいのである。もちろんそのときは，前の

全国アマチュアCG団体一覧 追加版

前回の掲載に間に合わなかった方々です。参加希望の方は，下記の連絡先に自己紹介を送ってください。詳しくは4月号をご覧ください。

❶ チーム名 ❷ 代表者名 ❸ 連絡先 ❹ 活動内容 ❺ 入会制限 ❻ 構成人数 ❼ その他

*

❶ 011 DōGA
❷ 清水拓詞（しみずひろし）
❸ 〒059-19北海道勇払郡追分町本町6丁目39番地
❹ 情報公開，CGA研究会，交歓会，その他……
❺ DōGA・CGAに関心のある，やる気のある方。男女，年齢問いません。
❻ 4人
❼ （現在の問題点より）CGAに関する資料不足，人材不足。何をしようか，どんなことができるかなどいろいろ考えてみましたが，何にも決まってません。私は，プログラムなぞBASICがちょっとできるくらいです。ちなみに私は，学生です。ヒマなのは私だけなのです。だからこの先どうなるかなんぞ，わかったもんじゃありません。できる限り頑張っていきますが，責任は持てませんのでそのへんをよろしく

❶ レッド・スターレッツ
❷ 松田英明
❸ 〒279千葉県浦安市堀江4-9-40モア415
❹ CGを用いた動画を作る（実はよく決まっていなかったりする）
❺ 暗いやつはだめ。女の子は水着審査並びに写真審査の上……ジョークです。アブナくなければOK！
❻ 2人
❼ いかんせん社会人なので暇がない。人がいない！　理屈なんて知らなくたっていいよっていってるのに。シナリオ作る人とかデザインする人とかでもよいと思うんですけどね

❶ Ari Assembly
❷ 本橋正成（もとはしまさなり）
❸ 〒153東京都目黒区祐天寺1-23-21
❹ 代表者が素人なので助け合っていきたい。月に1回ぐらいの茶会とIT-V1200を使ったNETを開く予定である。初心者，女性の方大歓迎！当然CGA，プログラム，音楽を中心に真面目に遊んでしまおう
❺ 特になし。気に入ったら62円切手同封の上，自己紹介を書いて郵送していただきたい。モデムがあると有利だ！
❻ 新設団体
❼ 1．代表者が素人（ただいま勉強中） 2．2月現在，X68000ACE-HDのHDがクラッシュ！（修理中） 3．ネットを開く時間以外のtelは私がとってしまう（研究中） 4．金がない……

❶ Tak-CGAチーム（仮称）
❷ 清野文雄
❸ 〒957新潟県新発田市古寺146
❹ 情報交換。および作品制作の作業分担など
❺ いまのところ，アクティブに情報を交換したいので，Tak-NET(BBS)会員の希望者で始めます（そのうち軌道にのってきたらオープン的にやりたい）
❻ 新設団体

❶ GINAX
❷ 三品徹（みしなとおる）
❸ 〒602京都市上京区御前西裏上ノ下立売上ル北町570-1
❹ 作品交換など
❺ 京都市に住んでいる人
❻ 2人
❼ いままで音楽の活動はやってたが，CGAはやっていない

❶ まだ考えていません（人が集まりしだい考

監督も１スタッフとして参加すべきだ。

今回の場合，とにかくMの好きにやらせることにした。“何がかっこいいか”ということもテーマになり得るし，もしMが本当にただかっこいいだけの作品を作り続けるなら，いずれ物足りないと感じる日がくるだろう。

翌日，Mが最初に現れたので，２人でもう少し具体的な内容を詰めてみる。本人の希望も考慮して，Mのオリジナル曲に合わせたイメージビデオみたいなものを制作してみないかと提案する。Mも異存がないという。各自が独自に作ったカットを集めて適当に編集するというやり方で，とりあえず作ってみようということになった。この方法だと，期間が限られている場合でもできあがったところまでで作品にすることができるほか，厳密な打ち合せや準備が必要なく，個々に独立制作できるメリットがある。しかし，画像としては脈絡がなく，作品としてのまとまりに欠けるのが多少心配だ。

独自に制作するといっても，出てくる物体だけでも統一しておくべきだろう。私は，「トロン」のような電子バイク，「ウイニングラン」のようなカーレース，「STAR WARS」のような宇宙戦闘もの，「グラディウス」のようなシューティングゲームの3D化という４つのパターンを考えた。特に最後のシューティングゲーム案は，敵キャラの形状デザインが楽しいし，最後にボスキャラを倒せば形だけでもオチがつき，まとまりがよいので美味しいと薦めてみた。が，Mは自分の曲のイメージからすると，電子バイクがよいというのでそれに決まった。

ちょうどMが自分の曲のテープを持ってきているというので，聞いてみることにした。確かにテンポがよい曲だ。が，はっきりしたメロディがなく，ゲームのBGMか間奏曲といった感じだ。脈絡のない映像と，メロディのない音楽では，映像と音楽どちらもメインになり難く，ピントのぼやけた作品になってしまう。多くの者から同様の指摘を受け，さすがにMも認めざるを得なかったが，ここはひとつなんとかMに頑張ってもらうことにした。

そのころになって，やっと新米３人組がやってきた。それぞれちゃんとアイデアを考えてきたというので，みんな期待して集まってきた。「あんまり寝ぼけたこというと，物が飛んでくるから気をつけや」と脅かす先輩たち。本当に投げつける物を手にする者までいる。

新米Ａ「私は３つ考えてきました」

先輩　「ひとりで３つとは偉いやんけ（別にヤクザではない。大阪弁である）」

新米Ａ「１つめのタイトルは“地上げ屋”」

先輩　「なんやねん。わけわからんがなぁ。まぁストーリーを聞かせてもらおう」

新米Ａ「貧しい家族が食事をしているところに，地上げ屋の団体が乗り込んできます」

先輩　「おいおい，それはCGか？　実写か？」

新米Ａ「CGです」

先輩　「貧しいという雰囲気を出すのも難しいが，CGで自然な人体を描くのはちょっと無理やで。“日本昔ばなし”よりもずっとデフォルメして，描くとまんがチックになるが，この話はコメディなのか？」

新米Ａ「シリアスです。地上げ屋は，いろいろ嫌がらせをするんですが，お父さんが頑張って，みんな追い払ってしまいます」

先輩　「それから？」

新米Ａ「翌朝，戦車がやってきて，それで終わりです」

この瞬間，いろんな物が投げつけられた。

新米Ａ「まぁ次のを聞いてください。２つめは“蟻の１日”です」

先輩　「今度はまともそうやな」

える）
❷　藤井信行
❸　〒640-04和歌山県海南市高津667
❹　参加者が何人かあれば（人数によって）活動内容が変わってくるので，人数がそろいしだい内容を考えていきたいと思います
❺　アマチュアの人。CGAに興味のある方で熱心な人。なるべく高校生以上の方歓迎します。できれば和歌山の人。大阪の人でも可能
❻　約５名
❼　活動歴は２年ぐらいで，まだ活動というほどのことはやっていないが，Ｃ言語を２人で勉強している最中です。あとひとりはＣ言語を使いこなしているようです。けっこう頻繁に活動しています。しかし，X68000を使いこなせる人が少なく，ゲーム以外のソフトが高額で手に入れにくいこと，人数の少ないこと，パソコンフレンドが近くにいないことなど問題点は多い
❶　**L.M.D.**
❷　平田剛
❸　〒761-07香川県木田郡三木町池戸769報徳苑103号
❹　DōGA・CGAシステムによるCGAの制作。
❺　香川医科大学まで来ていただける方で，高校または大学に在学中の方。コンピュータに関する知識のない方でも結構です（HD付きのX68000またはPC-98をお持ちの方，大歓迎です）
❻　２名
❼　1988年から活動を開始し，今年で３年目を迎えます。細々と活動を続け，今年のアマチュアCGAコンテストに初めての作品「Memory」を出品しました
❶　**“彩色映人”（さいろえいと）**
❷　上田晃好（うえだてるよし：RANA）
❸　〒457愛知県名古屋市南区西又兵エ町１ー９
・アクセスダイアル　052(614)0503（ネット専用24時間）
・プロトコル　N8IXN 2400MNP5
ID=“GUEST”（パスワードなし）
❹　名古屋地区を中心に，DōGAチーム専用連絡ネットを開設しています。DōGAからのデータなどを流すだけでなく，独自の企画を作り，プログラムやデータをみんなで楽しみながら共同制作していきたいと思います
❺　できればパソコン通信をやっていらっしゃる方，あるいはやろうという方
❻　約４名
❼　ただいま，タイトル文字の形状データとそのコンバータに取り組んでいます。また，CGAシステムの勉強会，コンテストのビデオや「３・６・５」などの上映会と配布も行っています。
❶　**電脳クラブ電影隊**
❷　辻村高
❸　〒810福岡市博多区博多駅前1-4-1シャープOAショールーム内「電脳クラブ」
❹　面白ければなんでも……
❺　老若男女問わず
❻　不定
❼　代表者自己紹介：九州地区のチーム紹介に，福岡がなかったのを見て思わず立候補しました。皆さんの積極的な参加をお待ちしています。近々クラブで主宰しているBBSにDōGAのボードを開設しようと思います
博多っこBBS　092-481-3520
❶　**コミュニケーション・アーツ68（CA68）**
❷　安藤優子
❸　〒813福岡市東区香住ヶ丘2-14-25エスポアール102号
❹　主にビジュアル制作。人に訴えかけ，なんらかの形でコミュニケーションしてゆくことが目標です。2D，3D，音楽などいろいろな手段を活用したいです
❺　時間さえあれば，連日連夜の徹夜で無償のチーム活動に積極参加できる人
❻　４名
❼　問題点：人手不足，資金不足，スイミン不足。「テクニック先行」より「全体の流れ」を重視したアニメーションを作成中

新米A「地下にアリの巣があって，女王アリの部屋やら，食料庫やら，たくさんの小部屋に分かれています」
先輩　「なかなかCG化しやすくていいじゃないか。それから？」
新米A「そこまでしか考えていません」
再び，物が飛んだ。
先輩　「それで，3つめはなんや」
新米A「3つめは“居酒屋の夜8時”」
先輩たちは何もいわず，投げる物を用意する。
新米A「ある男の視点から見た映像から始まります。夜の町を歩いていると，暗い通りに1軒の居酒屋から光が洩れています」
先輩　「映像的イメージがしっかりしているのがいいな」
新米A「そのまま入って行くと，おかみさんが“今晩は遅かったのね”と声を掛ける」
先輩　「何度もいうけど，雰囲気が出るほどの自然な人体は難しいで。まぁいい，それから？」
新米A「そこまでしか考えていません」
3度，物が舞う。
先輩に物を投げつけられても最後までいった新米Aは，なかなか根性がある。将来，DōGAを背負う大物になるだろう。初心者なのだから，マトはずれ的なアイデアになってしまうのは仕方ないだろうし，新米Aのいうようなことが簡単にできない現状のほうが間違っているのかもしれない。しかし，完成できそうにない作品企画をしても意味はない。やはり作品を完成させることが優先だ。
新米B「私は，戦闘機による空中戦をやってみようと思いました。戦闘機なら関節もないし，動きのデザインがしやすいのではないでしょうか」
先輩　「おー。なんかものすごく優等生的な答えだ。でも空中戦というのは，背景が何もないので，動いていることを表現しにくいで」
新米B「ええ，そこで，戦闘区域をどんどん変えていくのです。最初は海，それから林になって，山岳地帯へ。ラストのクライマックスはグランドキャニオンです」
先輩　「グランドキャニオンか。迫力のある映像ができるな。でも，短い作品でそんなに背景が変わっていくのは，ちょっと無理があるんじゃないか」
新米B「べつに最初からグランドキャニオンでもいいです。それから，最初から最後まで，ある戦闘機の視点の映像でまとめようと思うのですが」
先輩　「それはだめだ。そういう奇をてらった演出は失敗の元。そんな制限をつけてしまうと，できる映像が限られたものになって，見ているほうもすぐ飽きてしまう」
新米B「そうでしょうか。斬新で面白いと思いますが」
先輩　「斬新じゃない。使いものにならないから，誰もやっていないだけなんだ。ウソだと思うのなら，実際にそんな映像ばかり作ってみればいい。30秒間見るに耐えるものができれば誉めてやるよ。その場合でも，他の視点から見た映像をおり混ぜるほうが，ずっとよくなる」
これは実際に作品制作をしてきた経験がものをいう。
新米C「私の考えたテーマは“不法駐車”です」
先輩　「こりゃまたわけのわからんこといい出したなぁ」
新米C「問題になっている不法駐車を取り上げ，不法駐車によって事故が発生する過程を映像化するのです」
先輩　「それはシリアスなのか？」
新米C「もちろんシリアスです。ちょうど政府広報のようなイメージです」
先輩　「う～ん，政府広報のCG化というのは，アマチュアCGA始まって以来の試みだろうな」
新米C「車や人体の形は，非常に単純化すれば十分可能だと思いますが」
先輩　「技術的な問題点はない。発想も非常にユニークだ。だけど，そんなの作って何が面白いんだ？　Mは3人の意見を聞いてどう思う」
M　　「やはり，グランドキャニオンのドッグファイトがいちばんまともだと思います。曲にも合わせやすいし」
ということで，電子バイクの路線がドッグファイトに変更することで意見が一致した。同時に，“かっこいい”“スピード感”という2つのコンセプトが決まった。
あとは，スタッフ内で自由にアイデアを出し，監督(または取りまとめ役)が，皆の意見をまとめていく。

柚姫の明るい悩み相談室

はじめまして，柚姫です。“ゆず”と書いて“ゆう”と読みます。今度，私，2代目姫を襲名しました。で，気分一新，コーナー名も変わりましたが，“あき姫の迷える子羊のコーナー”同様にかわいがってください。このコーナーの内容はいままでどおり質問・お便り中心です。ま，気が向いたらちょっと違うことも……。
最初ですし，少々自己紹介など。今年大阪大学に入学した新入生です。DōGAに参加するきっかけは……気づいたらいたんですけど。ま，CGAで映画なんか作れるといいなぁ，なんて思って。実は，実は，コンピュータなんてこれまで触ったことないんです。だのに，だのに，いきなり原稿は書かされるし。来年の春にはCGAシステムがきっと，いやたぶん，いやもしかしたら，使えているはず……うるうる。けど，ご質問にはせーいっぱい（迷？）解答していきますのでよろしくお願いしま～す。

相談者：あの～，実は点光源の使い方なんですけど。3月号のあき姫のコーナーにあったとおりにやってみたのですが，どうもうまくいかないんです。私はどうしたらよいのでしょうか。
姫：そ，そうなんです。ごめんなさい。あき姫に代わってお詫びします。実は3月号の解説にミスがありました。目ざとい方はもうお気づきと思いますが，左右の括弧の数が一致しておりません。
（書式）
誤　{mov(X Y Z) light point(rgb(R G B) L}
正　{mov(x y z) light point(rgb(R G B) L)}
当然，（例）も間違っているわけですね。さあ，3月号と赤ペンを取り出して添削しちゃいましょう。
（例）
誤　{mov(10 20 30) light point(rgb(0.5 0.5 0.5) 200}
正　{mov(10 20 30) light point(rgb(0.5 0.5 0.5) 200)}
というわけです。本当にごめんなさい。
相談者：私は“名なしのごんべえ”です。CGAシステムを申し込むのに，自分の住所も名前も書くのを忘れ，Oh!Xの連載の各読者通達事項のコーナーで全国指名手配されました。スタッフの皆さんお手数をおかけして申し訳ありませんでした。こんな私ですが，どうしたらよいでしょうか。
柚姫：そ～いうことは，お早めにご連絡ください。ちゃんと送ります。でも，あの～，名なしのごんべえさん。今回のお便りにもお名前，住所が書かれていないんですけど……。
面白いお便りから，CGAシステム・DōGAプロジェクト・その他についての質問やお叱り，ご要望といった真面目なお便りまで何でもOK。どんどんお便りください。

「具体的に飛行機は何にするの？」
「サンプルデータのF16を流用すればいい」
「飛行機だけじゃつまらないから，陸上部隊も出そう」
「ややこしくなるから反対。敵味方の区別がつかないぞ」
「味方の兵器はすべて赤い色で統一しては？」
「赤いF16なんて見たくないよう」
「敵の戦闘機の数はどのくらい？　5機ぐらいかな」
「20機ぐらいの編隊も，かっこよさそうだけど，動きのデザインが面倒だな」
「機数はなんで戦闘をしているかの問題じゃないの？」
「短編なんだから，設定なんてどうでもいいと思うけど」
「F16に限らず，いろんな機種を出したら？」
「それも，ややこしくなりそう」
「でも，複葉機対F16なんて面白そう」
「あっそのアイデアいい。複葉機がF16をかたっぱしから撃ち落としていくんだ」
「F16と複葉機じゃ複葉機に勝ち目なんかないじゃんか」
「そこがナンセンスコメディなんだよ」
「おいおい，いつからコメディになったんだよ。かっこいい，スピード感じゃなかったのかい」
このように，仲間同士うだうだ話し合っているうちに，いろいろアイデアが出てくる。監督が新しいアイデアに関心を示せば，元の内容など気にせずどんどん話を膨ら

寺田の教育的指導

「お客さん，えらい久しぶりやねぇ！　へぇー，5ヵ月ぶりになるか。長いこと，どこ行っとったんや？え，高野山にこもって修行しとった？　うーん，そのわりには，ちょっとも文章，上手なっとらんな……」。というわけで（?），1年たったこの連載ですが，心も機も一転せず，相変わらずのこのコーナーの始まりです。さっそく本文へ突入だっ。

今回ご紹介する作品は，本職の日本画家（!）でいらっしゃる東京都の森山知己さん制作の“走る「MASAYA」”と“メリーゴーランド（特にタイトルはついていませんでした）”です。これは現在制作中の作品の一部で，完成すればMASAYA君がCRT上でおもちゃと遊び回るという作品になるそうです（これは楽しみ！　次回のCGAコンテストの入選有力候補！）。

モデリングのセンス，色の配置などさすがプロといった感じで，独特の可愛く楽しい雰囲気のあるものに仕上がっています。CGAコンテスト入選作の「デファイナブル・ファンクション」の色の使い方にもショックを受けましたが，この作品も新鮮な気持ちよさを感じました。いままでのCG作品は，どちらかというとメカニカルなものが多かったですが，これからはこういう作品も増えていくことでしょう。

技術的な面では，この作品はこれまでの連載で解説してきた基本的なテクニックがすべてうまく使いこなされています。皆さんも，CGAシステムをこの程度にまで使えるように「精進」してください。まずモデリング，これは作る人のセンスに左右されますが，比較的簡単な立体の組み合わせでうまく処理しているところは参考になるでしょう（人体はちょっと違いますが）。私は木馬の目が可愛くて気に入ってしまいました。

次に，色・アトリビュートですが，台の部分の赤色と，木馬の白の組み合わせがとても鮮やかできれいですね。メインの色があり，それが引き立つように周りの色が選ばれている気がします（このへんは，さすがプロの画家さんですね）。

物体の動きも，MASAYA君の手足の動き，回転しながら上下する木馬など，構造体を使ってうまく表現されています。どうやったらこういう動きができるか皆さんも考えてみてください（FSCに慣れるには，いろいろとやってみるのがいちばんです。高校程度の数学の勉強もしてください）。

また，これとは別に「少し凝った」形状ということで，“人体モデル”も送ってくださいました。しかし，これは「少し」どころではなくかなり凝った形状データで，人体らしい特徴がよく出ています。面数が多いのは，人体という題材の都合上仕方ないですが……。

＊

ところで，森山さんからいくつか質問が届いていますので，この場でお答えしたいと思います。

Q：RENDの実行のとき，データの数が多いと，コマンドラインに収まらないのですが。

A：COMMAND.Xの制限で，255文字以上のコマンドは実行できません。たくさんの物体を使って作画するとき，RENDに与える引数が多くなってこの制限に引っ掛かってしまいます。こういうときは形状ファイルや，アトリビュートファイルをいくつかまとめてひとつのファイルにしてください（ひとつのファイルに2つ以上の形状データが含まれていても，ちゃんとRENDは受け付けてくれます）。

多少面倒ですが，エディタを使うのが確実です。たとえば，A.SUFとB.SUFというファイルをひとつにまとめたいときは，まずエディタ（ED.X）を立ち上げ，A.SUFを読み込みます。そして，F2キーなどで，ファイルのいちばん最後の行より下の行へカーソルを持っていき，そこでESC-YコマンドでB.SUFファイルを後ろに読み込んでください。ESC-Tコマンドでファイル名を変更してから（拡張子はSUFとしてください），セーブすれば作業は完了です。

3つ以上のファイルのときも同様のことを繰り返してください。できあがったファイルは，そのままRENDにかけることができます。

Q：FFのオプションスイッチの使い方がよくわからないのですが。

A：森山さんのように，データが多く複雑な動きをするものを作るようになると，「メモリが不足しています（制御用）」とか，「メモリが不足しています（データ用）」などのエラーがすぐ出るようになります。読んで字のごとくワークエリアが不足していますので，オプションスイッチで増やしてください。

-scで制御用バッファ，-sdでデータ領域の大きさを指定できます。普通は，-sc10000 -sd100ぐらいで問題なく動作すると思います。足りなければ，もっと数字を増やしてください。また，デフォルトでは読み込めるフレームソースの大きさは，32Kバイトまでなので，それ以上に大きいときは，-sfオプションで大きさを指定してください。

あ，それとPESから使うときは，「オプション」のウィンドウは使わずに「フレームフィルタ」ウィンドウに「-sc10000」などと書いてください。RENDのバグ，じゃなくて仕様のせいで2文字のオプションは「オプション」ウィンドウからうまく使えないのです（ご了承ください）。

＊

さらに，森山さんからは非常にていねいなアンケートの回答と，プログラムの改良点，今後のCGAのあり方などに関する意見，そのうえビール券まで送って頂きました（あまりDōGAにエサを与えないでください。調子に乗ってしまいます）。そのお礼というわけでもないのですが，森山さんにはCGAコンテストの作品集ビデオをお送りしたいと思います。

ついでに，今回は私の作ったプログラムBOMB.Xを紹介します（この号が出るころには，ネットにもアップしているはずです）。このプログラムは，物体が爆発して飛び散っていくところを表現したくて作ったもので，「ボイジャー」やCGAコンテストのオープニングアニメにも一部使われています。

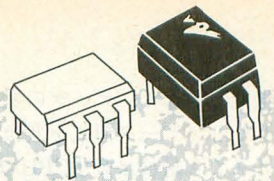
アルゴリズムは簡単で，まず物体データをバラバラにして，物体の1面1面を独立した形状データとして切り離します（つまり100個の面からできた物体なら，100個の「部品」に分かれるということ）。そして1つひとつの部品が飛び散っていく動きを計算して，それをフレームソースの形で出力します。それをFFにかけて，RENDでレンダリングすれば，物体が飛び散るアニメのできあがりです。写真を見てもらえば，何をやっているかよくわかると思います。

CGAシステムはデータの規格がきっちり決まっているし，FFやRENDといったしっかりしたプログラムがすでにあるので，このような安易なアルゴリズムのプログラムでも，結構面白いことができてしまいます。皆さんもいろいろと挑戦してみてください。私もあと2つ3つ「小ネタ」を考えているので，できあがったらまた発表したいと思います。

最近は送られてくる作品のレベルが上がってきて喜ばしい限りですが，少し数が減ってきています。作ってる途中で行き詰まってしまったものや，こういう事をするにはどうしたらいいのか？　といった質問もお待ちしています（そのほうが解説もしやすいですし，なんちゃって）。それではまた次回お会いしましょう。さようなら。

ませよう。要は監督が気に入るかどうかであり，スタッフがアイデアをたくさん出し，監督が取捨選択しながらイメージを固めていけばよいのだ。
監督「基本的なストーリーは，最初，複葉機がグランドキャニオンをのんびり飛んでいる。そこにF16の編隊が現れ，攻撃を仕掛けてくる。複葉機が反撃を仕掛けると，これがめちゃくちゃ強い。F16を片づけた複葉機が，またのんびりと飛び去っていく（起承転結の構成がちゃんとできている点に注意）」
「見せ場は，ドッグファイトってことになるんだろうけど，複葉機はどうやって戦うんだろう？」
「プロペラの間からマシンガンを打つんじゃないのか」
監督「ちょっとまとも過ぎ。もうちょっと過激でいい」
「ミサイル？」
監督「ミサイルはF16。複葉機は別の方法で戦わないと」
「ちなみにF16のミサイルは熱線追尾型だから，プロペラ機には通用しません。エンジンの熱量が足りないんだ」
監督「事実なんて映像作品なんだからどうでもいい。F16はミサイルをたくさん発射しないと絵にならない」
「複葉機から出ているタイヤで蹴飛ばすとか」
監督「面白いけど，いまいち」
「横になって，2枚の翼でF16の翼を切り落としたり」
監督「むちゃくちゃだけど，そんなのもいいな。F16の方は5機ぐらいあって，初めの1，2機はけっこうまともな方法でやっつけるんだけど，だんだんむちゃくちゃになってきて，最後の1機は見ている人が“ウソつけ！”と怒鳴りたくなるほどエスカレートしていくんだ」
「複葉機が変形して，戦闘ロボットになるとか」
「なんだよ，アニパロかよ。オレも好きだけど」
監督「その路線はやらない」
「変形して，袴，ちょんまげ姿になったらうけるで」
監督「やらないものはやらない」
「ロボットとはいかなくても，変形して最新型の戦闘機になってもいいと思うけど」
監督「時代遅れの複葉機がF16を倒すことに意味があるので，最新メカになってはいけないんだ」
さて，作品企画も最終段階になってくると，監督がだんだんガンコになってくる。それは，監督の頭の中で自分のイメージがしっかりできた証拠であり，自信が出てきたことを意味する。“この監督は私のアイデアを取り入れてくれない”などとスネずに監督のつかみ始めたイメージを積極的に理解し，まとめる方向に持っていこう。
監督「オープニングは，青空のグランドキャニオン。遠くのほうから複葉機が飛んでくる。ふらふらと，いかにもオンボロって感じ。ロングの長回しで（長回しは1カットが長いこと）。そしてタイトル。そのあとBGMの雰囲気が激しくなってくる。空の片隅に何かが現れる。なんだ？　と思わせて，いきなりF16のアップ。ポンッポンッと短いインサートカットを入れて，緊迫感をあおる……」
最後に監督から作品の解説をしてもらう。どんなストーリーということだけではなく，各シーンがどのようなイメージなのかということが重要となる。全スタッフが，監督のいわんとするイメージを共有しなければいけないのだ。
以上が作品企画会議のようすです。このあと，監督は担当者といっしょに絵コンテの作成に入り，他のスタッフは，資料集め，形状データのデザインなどを始めます。監督は，自分のイメージをしっかり固めて，全カットの細部まで決定していかなければいけないのですが，アイデアに詰まったりすれば，何度でも企画会議を開くことが必要です。なお，この作品の制作過程については（ちゃんと制作が進めば）また次回にでも報告しますのでお楽しみに。

おわりに

最後にもうひとつ企画に大切なコツの話で締めくくりましょう。それは，“365日，24時間企画をする”ことです。いきなり“さぁ作品企画をしよう”といって思いつくなんてものではないので，常日頃考えておくのが大切なのです。とはいっても，いろいろ忙しい毎日，作品企画ばかり考えているわけにもいきません。ですから，思考の5％ぐらいだけを常に企画に回しておいて，何を見，何を聞き，何をさわっても，一度“これは作品になるんじゃないか”と想像してみるのです。朝，食卓について箸を手に取ったら“箸の形って，CADで作りやすいな”と気がつきます。そして“食卓ってのはいろんなキャラクタが集まった集会場だな”と見回すのです。“つま楊子でフェンシングなんて楽しそうだな”と，赤と青の箸がつま楊子でフェンシングをしている映像が目に浮かべば完璧です。
何もその場でストーリーを作れというのではなく，アイデアの蓄積が大切なのです。試しに明日の朝，家を出て学校（あるいは会社）に着くまでの間，“これは作品になるんじゃないか”を試してください。ネタの1つや2つ必ず見つかるでしょう。
作品企画は，作品制作の中でも楽しく，盛り上がるところです。あまり難しく考えずに，積極的に楽しむのも大切でしょう。今回解説した企画法はあくまで，これをヒントにしてくださいといった程度のものです。これを読んで，皆さんも作品を制作する気になっていただければ幸いです。
さて，「FAR SIDE MOON」のウイルス事件ですが，当スタッフも購入した者がいたため，一時期すべてのディスクの持ち出し，発送を禁じるの騒ぎが起きてしまいました。しかし，ウイルスを確実に見つける方法も確立していない現状では，当チームも同様のトラブルを起こさないという保証はありません。憎むべきはウイルスを作った犯人であり，ワクチンソフトの無償配布など多大な損害を出したアートディンクは最大の被害者だと思います。悪意を持ったウイルスによって，「FAR SIDE MOON」の売り上げが落ちたり，アートディンクが経営の危機に陥ったとしたら，私たちユーザーにとっても不幸であり，ある意味でウイルスに屈したことになるような気がします。ですから，アートディンクからワクチンを受け取ったユーザーは，実費だけでもアートディンクにカンパしてはいかがでしょうか。

ハードウェア工作入門《1》

基本インタフェイス回路 その1

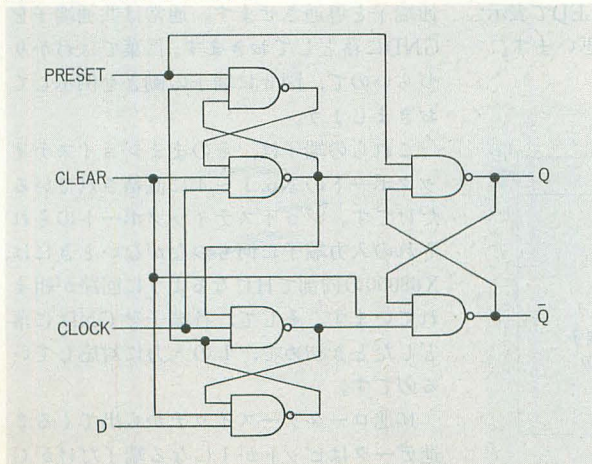
Misawa Kazuhiko
三沢 和彦

いよいよ今月からハードウェア工作入門の始まりです。理論編，工作編，活用編の3回で1つのテーマを扱っていきます。最初は部品の少ない簡単な工作でI/Oの基礎から解説していきますので，ぜひとも続けて読んでいってください。

先月号の予告編でお伝えしたように，さっそく自作ハードを X68000 のジョイスティックポートにつなぐことを考えてみましょう。まず初めに，どうしてジョイスティックポートにつなぐと外部機器とデータのやりとりができるのか皆さんは理解していますか？ それは，ジョイスティックポートがI/O(Input/Output) インタフェイスの役割を果たしているからです。今回は，連載第1回ということで，まず，インタフェイスの概念とその仕組みというもっとも基本的な事柄から説明していきたいと思います。

インタフェイスの基礎とジョイスティックポート

そもそも「インタフェイス」という言葉は「境界面」という意味で，コンピュータの世界では，異なる2つのハードウェアを仲介してつなぐ装置のことをいいます。特にコンピュータ本体と外部機器とを接続するのに使われる場合をいうことが多いようですが，基本的には，キーボードやディスプレイといった入出力機器とCPUとをつなぐための内部回路はすべてインタフェイスというべきです。

このインタフェイスの重要な役割は，

1) データを受け渡しする機器どうしの間で互いにやりとりができるようにデータ形式を変換してやる。

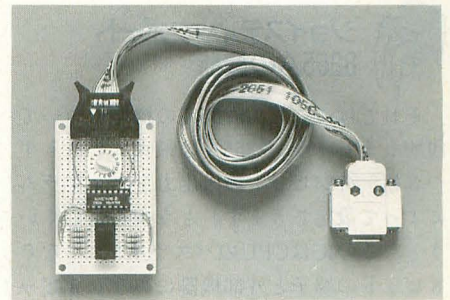
2) 時間で変動していくデータを処理できるように一時的に保持してやる。

の2つです。これらの点について，少し具体的に突っ込んでみましょう。その前にコンピュータ内部でのデータ形式について復習しておきます。

データは基本的に1（真）か0（偽）で表し，これを1ビットといいますが，X68000は16ビットまとめて処理します。ただし，インタフェイス回路は通常8ビット単位で処理します。8ビット（16ビット）まとめてという場合には，信号線が8本（16本）横に並んでいて，それぞれの信号線上に各ビットのデータが乗っていると考えればよいのです。このような横並びのデータ形式を「パラレル」データといいます。

コンピュータ内部ではこの横並びの信号線をバスラインといい，CPUからメモリ，そしてキーボード，CRTディスプレイ，ディスクドライブなどの各回路につながっていて，隅々までデータを流します。このとき，たとえ1/0の2進データをそのままCRT に送ったとしても，画面上に文字は表示できません。画面に文字を表示するには，ASCIIコードのような2進データをCRT用のRGB信号に変換してやってからCRT に送らなければなりません。

図1 D-フリップフロップ

また，キーボードからの入力にしても，我々が入力するデータは各々の文字に対応するキーを押すだけで，キーの押された場所に対応してそれぞれ2進の文字コードデータに変換してやらなければCPUは処理できません。これらがすなわち，1)でいうデータ変換の意味です。

製作するインタフェイス回路

次に，キーボードから文字が入力され，文字コードに変換されてバスラインにデータが乗ったとしましょう。このとき，CPUは何をしているのでしょうか。

CPUはたえずデータをやりとりしたり，あるいはCPU内部で演算している最中だったりするかもしれません。すると，CPUにしてみれば，いつキーボードからの入力があるかわからない状態で，突然文字データが送られてきても，読み過ごしてしまうのです。

そこで，2)でいうように，送られてきたデータを一時保持しておいて，CPUが都合のよいときにおもむろに読み出してくるようにしておく必要があるわけです。また，外部機器にデータを出力するときも，外部機器の読み取りタイミングに合わせるためにデータの保持が必要なときが多くあります。

さて，どんな入出力データでも，CPUが処理するのがパラレルデータである限り，インタフェイスは基本的にパラレルデータを扱わなければなりません。このようにパラレルデータと他のデータ形式とを仲介するインタフェイスをパラレルインタフェイスといい，このパラレルインタフェイスこそがインタフェイスのもっとも基本的な形だといえます。

パラレルインタフェイスの実際の回路は，図1に示すD-フリップフロップをバスラインのビット数分だけ横並びにしたものが基本です。この回路は，データの保持だけの

機能しかなく，データの変換は行えません。しかし，データ形式を変換するインタフェイスでも最後は必ずCPUのためにデータを保持しておかなければならないので，すべてのインタフェイスがこの部分を持っています。それだけ基本的なインタフェイスですから，専用のICにもなっています。

もちろんX68000の中でもパラレルインタフェイスはあらゆるところで活躍していますが，その代表的なICが8255AというICなのです。そして，問題のジョイスティックポートも，この8255Aそのものなのです。

ジョイスティックポート 8255Aの機能

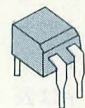
それではここで，8255Aの機能について説明しましょう。8255Aは最大８ビットずつ３組，計24ビットの入出力をプログラムによって設定できます。もう少し詳しくいうと，このICはCPUのバスラインにつなぐ８ビットの端子と外部機器につなぐ８ビットの入出力信号端子（ポート）を３組持っていて，このポートをそれぞれ入力として使うか出力として使うかを外部からの命令によって指定でき，このIC１個使うだけで３組のインタフェイスとして機能させられるわけです。

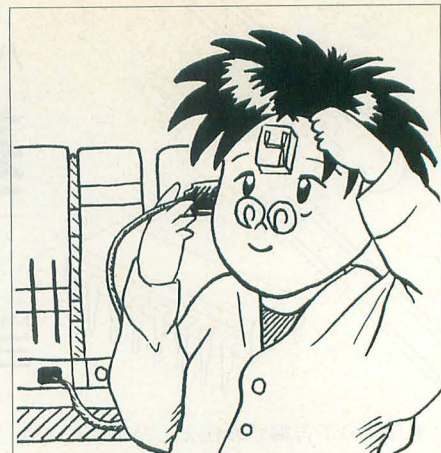
実際の信号の入出力は簡単で，入力にはポートの端子に5VをつなげばH，GNDレベル（0V）をつなげばLのデータとして，CPUが取り込みます。出力は逆に，CPUがHのデータを送ると端子に5Vが，Lには０Vが出てきます。そして，このインタフェイスとしての機能には，３種類のモードがあるのですが，初心者のうちは，これから説明するモード０のみ理解しておけばよいでしょう。ジョイスティックポートもこのモード０で使っています。

モード０では，ポートＡ，ＢおよびＣの半分ずつを任意にそれぞれ入力または出力ポートとして設定できます。そして，入力を指定したときはその入力時にＡ，Ｂ，Ｃの各ポートに加わっているデータがCPUに読み込まれます。ただしこのとき，ポートのデータは保持されないので注意が必要です。それに対し，出力を設定したときには，CPUから各ポートに出力されたデータは保持され，次にデータが出力されるまでは同じデータが出力され続けます。

表１にジョイスティックポートの各端子の信号の内容を示します。これは，X68000の取扱説明書の表をそのまま転載したものです。備考欄を見てわかるとおり，ジョイスティックポートの信号は＋5VとGND以外は，単に8255Aのポートが直結されているだけです。ジョイスティックポートの端子No.１〜4はポートＡを入力に指定して，４ビットの入力専用，No.６〜8はポートＣを出力に指定しての３ビット出力専用のインタフェイスになっています。

なお，端子No.６と７とは入力出力両用になっていますが，実際はCPUからポートＣにデータを出力してしまえば，そのデータが保持されて，端子No.６と７とに出力され続けます。ただ，端子No.６と７の出力は，端子４の出力と反転しているので，注意が必要です。このあたりの事情は，プログラムを組むときに問題になるので，再来月にX68000を実際に使って説明することにします。

以上の説明でインタフェイスの基本的な考え方となぜジョイスティックポートが，インタフェイスとして使えるかが理解できたと思います。それでは，次に今回の回路を挙げ，ハードウェアとして実際にパラレルインタフェイスをどう使っていくかを説明していきたいと思います。

LED＆SW 基本インタフェイス回路

今回のテーマは基本的な入出力ということで，外付けのスイッチからの入力を取り込む回路とポートからの出力をLEDで表示させる回路とを作ってみようと思います。

入力は４ビット，出力は３ビットなので，入力については０〜15の数字をロータリースイッチという回転式のスイッチからひとつ選んで，それを２進データに変換して取り込む回路とし，出力については０〜７の10進数をひと昔前の電卓などに使われていた７セグメントのLEDというものに表示させる回路とします。

実際の回路は図２のとおりですが，なんとICは１個しかないのです。最初の回路としては，部品点数が少ないほうが回路を理解しやすいし，工作も楽だと思います。それでは，入力部と出力部とに分けてそれぞれ詳しく説明していきます。

●入力部

まず入力部ですが，これは16進ロータリースイッチ１個だけという，なんとも工夫のしようがない回路です。それだけにスイッチの選択を誤ると，予想どおりの動作はしてくれないので注意が必要です。この16進ロータリースイッチには，５つの端子がついていて，そのうち１つが共通端子で，あとは２進４ビットの端子になっています。スイッチを０〜15の16通りを選ぶと，選んだ数を４ビットの２進数にスイッチのなかで変換し，ビットの１になる位の端子を共通端子と導通させます。通常は共通端子をGNDに落としておきます。言葉ではわかりづらいので，図３に端子の働きを例示しておきましょう。

これらの端子は，そのままジョイスティックポートのNo.１〜４に直結されているだけです。ジョイスティックポートのそれぞれの入力端子に何もつながないときには，X68000の内部でHになるように回路が組まれています。そして，各端子をGNDに落としたとき初めて，Lの入力に対応しているのです。

16進ロータリースイッチから出てくる２進データはビットが１になる端子だけがG

表１　ジョイスティック用コネクタ（アタリ社規格準拠）

端子No.	信号名	I/O	備　考
1	IOA0	In	8255のPA0端子
2	IOA1	〃	〃　PA1　〃
3	IOA2	〃	〃　PA2　〃
4	IOA3	〃	〃　PA3　〃
5	Vcc1	Out	＋5V
6	IOA5	I/O	8255のPA5/PC6端子
7	IOA6	I/O	〃　PA6/PC7　〃
8	IOC4	Out	〃　PC4端子
9	GND	—	

NDにつながり，ビットが0になる端子はどこにもつながらないようになっているので，ロータリースイッチから出てくる2進データがそのままジョイスティックポートに取り込まれることになるのです。

このとき注意しなくてはならないのは，各ビットの1のデータが入力のL，0のデータがHに対応している点です。このように正負が反転しているデータを正しく読み取るのはソフトウェアのほうで対処しますが，詳しいことは再来月のプログラム実習編で解説します。

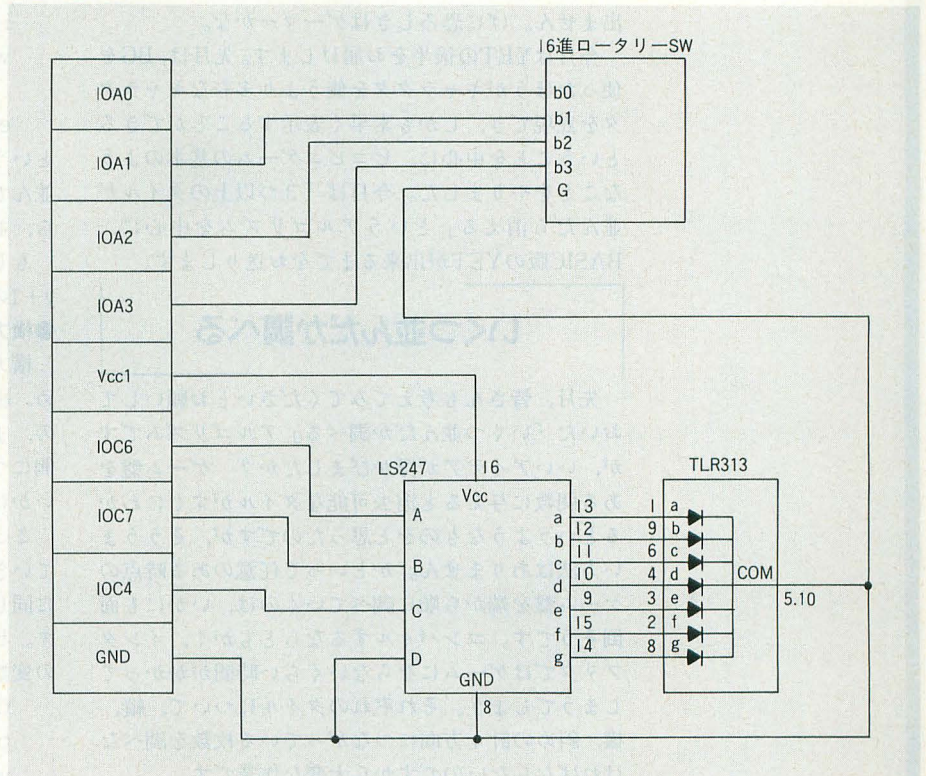
●出力部

出力部には，IC（LS247）1個と表示用の7セグメントLED（TLR313）1個とが使われています。7セグメントLEDは，図4のように，7本の線状LEDが「日」の字型に並べられているものです。この7本の組み合わせで10進数の0から9までを表現します。

ジョイスティックポートからは2進3ビットのデータが出力されてきますが，これをそのまま7セグメントLEDに入れてやってもだめで，7セグメントLEDに表示させるためにはデータを7本のセグメントの組み合わせに変換し直さなければなりません。LS247というICは，2進4ビットのデータを10進数に変換し，さらに，その10進数を7セグメントLEDに表示させる専用のICです。最初に述べたインタフェイスの役割のうち，出力のためのデータ変換を一手に引受けているのがこのLS247ということになります。そして，ポートの出力をそのままLS247の入力につなぎ，LS247の出力をそのままTLR313の各ピンにつなぐだけで簡単に表示部が実現されます。

ところで，LS247とTLR313との間に抵抗が入るのですが（図路図では省略），これは，LEDに大きな電流が流れすぎて壊さないように保護するものです。

＊

以上で，回路図の説明は終わりです。意外と簡単だと思いませんか？　まだわかりづらいという皆さんにも，来月の製作実習編，再来月のプログラム実習編で，まったく同じ回路について実際に動作を確認しながらくどいほど繰り返し説明を加えていきますので，ご安心を。次回の製作編では，部品表と実体配線図を載せて，初心者でも完成できるように丁寧に解説します。それまでに工具を揃えて待っていてください。では，来月。

参考　X1のジョイスティックポート

X1のジョイスティックポートにはPSG(Programable Sound Generator)の汎用I/Oポートを使っています。このポートもプログラムで入出力の切り替えができますが，ひとつのジョイスティックポートの中で，入力と出力とを使い分けることはできません。ですから，ふたつあるポートを両方使って入力と出力とを別々に設定してやらなければならないのです。

また，X1のジョイスティックポートには，電源の5Vの端子が出ていないので，X1のふたを開けて中からとりださなければなりません。これらの点に注意すれば，この連載に出てくる回路はそのまま流用できます。回路についてはX1本体とつなぐコネクタ部分だけ工夫すればよいので，X1ユーザーの皆さんは各自改造してみてください。

図2　回路図

図3　スイッチ

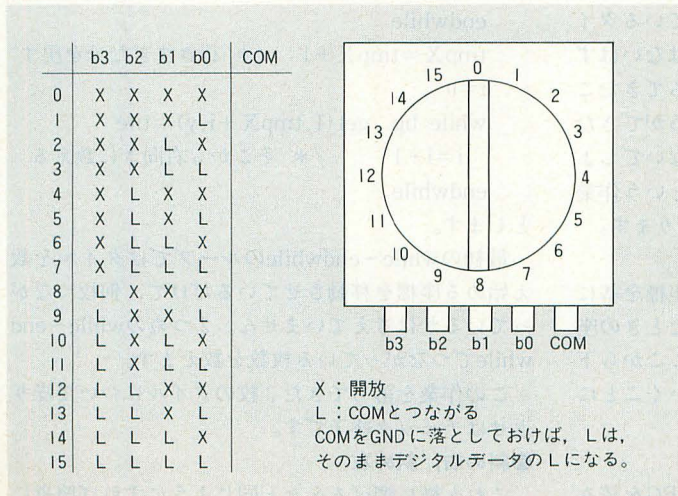
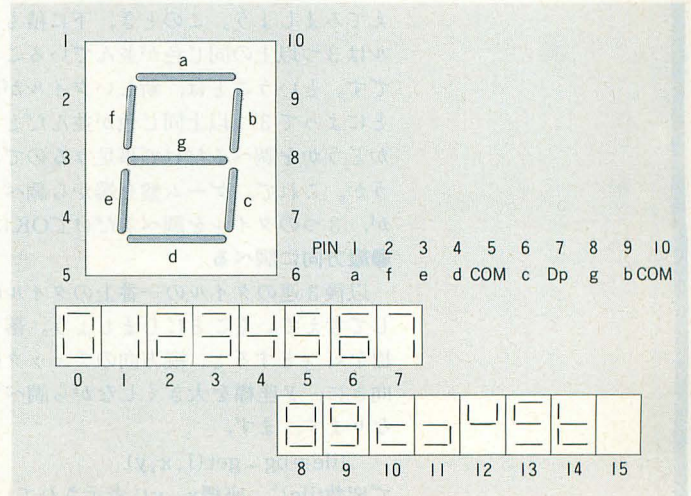
	b3	b2	b1	b0	COM
0	X	X	X	X	
1	X	X	X	L	
2	X	X	L	X	
3	X	X	L	L	
4	X	L	X	X	
5	X	L	X	L	
6	X	L	L	X	
7	X	L	L	L	
8	L	X	X	X	
9	L	X	X	L	
10	L	X	L	X	
11	L	X	L	L	
12	L	L	X	X	
13	L	L	X	L	
14	L	L	L	X	
15	L	L	L	L	

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

b3 b2 b1 b0 COM

X：開放
L：COMとつながる
COMをGNDに落としておけば，Lは，そのままデジタルデータのLになる。

図4　LEDの表示

BASIC

パズルゲームを作る(後)

Izumi Daisuke 泉 大介

さあ，Yet Another Columnの後編です。揃ったブロックを判断して消すアルゴリズムを考えてみましょう。また，ディスク版はコンパイラ版でしたが，インタプリタのままでも十分なゲーム性を持たせるための工夫もポイントです。

先月号の付録ディスクでお届けしたYETはいかがですか，楽しんでいただけていますか。編集室ではさらにスコアが伸び，現在の最高得点は（よ）嬢の42315点です。もうこうなると作者の私も手も足も出ません。げに恐ろしきはゲーマーかな。

今月はYETの後半をお届けします。先月は，BGを使ったほうがキャラクタを使うより多彩なキャラクタを表現でき，しかも素早く表示することができるということを中心に，ピコピコゲームの基本のようなことをやりました。今月は「3つ以上のタイルが並んだら消える」というアルゴリズムを中心に，BASIC版のYETが出来るまでをお送りします。

いくつ並んだか調べる

先月，皆さんも考えてみてくださいとお願いしておいた「いくつ並んだか調べる」アルゴリズムですが，いいアイデアが浮かびましたか？　ゲーム盤をある関数に与えると消去可能なタイルがすぐにわかるというようなものをと思ったのですが，そううまい方法はありません。かといって任意のある時点のゲーム盤を端から順に調べていくのは，いかにも面倒そうです。コンパイルするならともかく，インタプリタではゲームにならないくらい時間がかかってしまうでしょう。それぞれのタイルについて，縦，横，斜めの計4方向につながっている枚数を調べなければならないのですから大変な作業です。

そこでタイルが上から落ちてくるときの様子を考えてみましょう。このとき，下に積もっているタイルは3つ以上の同じ色が並んでいることはないはずです。ということは，新しいタイルが落ちてきたことによって3つ以上同じ色が並んだところができたかどうかを調べるだけで事足りるのではないでしょうか。これで，ゲーム盤を端から調べるという作業が，3つのタイルを調べるだけでOKになります。

●縦方向に調べる

以後3連のタイルの一番上のタイルの座標を基にして考えていくことにしましょう。落ちたときの座標をx，yとすると，縦方向のチェックはここから下向きに，y座標を大きくしながら調べていくことになります。まず，

```
tile=bg_get(1,x,y)
```

で変数tileに，座標x，yに表示されているBGを読み出します。ここで読み出されるのは，256×パレットブロック+1というコードです。これはbg_putでパターンを指定するときのコードと同じです。同じ色のタイルなら，コードは同じはずですから，

```
i=0
while tile=bg_get(1,x,y+i)
  i=i+1
endwhile
```

というループを実行すれば，同じ色のタイルが何個並んでいるかが変数iに求まります。iが3以上なら，並んでいるタイルを消せばいいのです。

もし3つ以上並んでいないなら，今度はtileにx，y+1のタイルをセットして下向きに調べます。

●横方向に調べる

横方向は左と右の両方を調べなければならないため，縦方向ほど簡単にはできません。とはいうものの，まず左側につながっている個数を調べ，次に右側につながっている個数を調べて加えるというのはいかにも間抜けな気がします。

ここで採用した方法はまずどちらか一方へたどっていき，タイルが違う色になったらそこから逆方向に同じ色のタイルの数を数えていこうという方法です。具体的にはまず一時的なx座標を保持するための変数tmpXを用意し，

```
tile=bg_get(1,x,y)
tmpX=x
while bg_get(1,tmpX,y)=tile
  tmpX=tmpX-1 /* まず左側に移動する
endwhile
tmpX=tmpX+1   /* 行きすぎた分を戻す
i=0
while bg_get(1,tmpX+i,y)=tile
  i=i+1       /* そこから右向きに数える
endwhile
```

とします。

最初のwhile～endwhileのループではタイルを数え始める座標を移動させているだけで，何枚つながっているかは考えていません。2つめのwhile～endwhileでつながっている枚数を数えます。

この作業を落ちてきた3枚のタイルについて繰り返せばチェック終了です。

●斜め右，斜め左

これも横に調べるときと同じようにすれば簡単に

チェックできます。左斜め上の方向を調べるなら,

```
tile=bg_get(1,x,y)
tmpX=x : tmpY=y         /* xとyの2つを
                           用意
while bg_get(1,tmpX,tmpY)=tile
  tmpX=tmpX-1           /* 左斜め上に
  tmpY=tmpY-1           /* 動かしていく
endwhile
tmpX=tmpX+1             /* 行きすぎた分を
tmpY=tmpY+1             /* 修正する
i=0
while bg_get(1,tmpX+i,tmpY+i)=tile
  i=i+1           /* 右斜め下に動かしていく
endwhile
```

でOKですね。左斜め上から右斜め下へ同じ色のタイルが何枚つながっているかをこれで知ることができます。3連のタイルそれぞれについて繰り返すことになります。あとは右斜め上の場合をリストを見る前に考えてみてください。

枚数チェックの汎用化

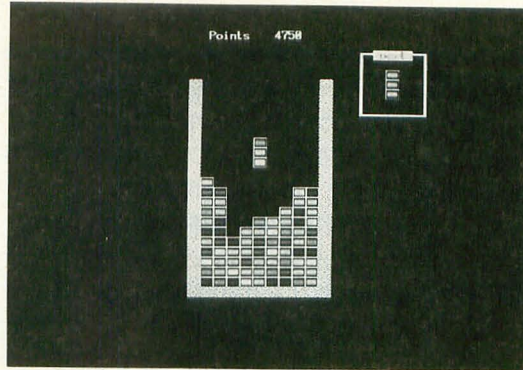
ここで紹介した方法は上から落ちてくる3連のタイルしか考慮していません。タイルが消去された場合,その上に積もっていたタイルは自由落下し,その結果新たに3つ以上のタイルが並ぶとこれもまた消去されます。これら2つの状況は非常によく似ています。うまく同じ方法で処理できるようにならないでしょうか。

3連のタイルを,3枚のバラバラのタイルが「たまたま」塊になって落ちてきたのだと考えてみましょう。たまたま塊になって落ちてきただけですから,一番上のタイルの座標を基準にほかのタイルの色のつながりを調べようという上の方法は訂正しなければなりません。タイルの座標を1枚ごとに配列に保存し,枚数チェックはこれを基に行うことにしましょう。

具体的には次のような要領です。配列名をdoubtとしておきましょう。3連のタイルが下に落ちたら,

```
doubt(0,0)=x : doubt(0,1)=y
doubt(1,0)=x : doubt(1,1)=y+1
doubt(2,0)=x : doubt(2,1)=y+2
```

と,3つのタイルの座標を配列に収めます。このとき,doubtEnd=2として最後のタイルの座標が入っている場所を表しておきます。同色のタイルが3つ以上並んでいるかどうかのチェックは,

```
for t=0 to doubtEnd
  tmpX=doubt(t,0)
  tmpY=doubt(t,1)
  tile=bg_get(1,tmpX,tmpY)
  /* 縦のチェック
  i=0
  while tile=bg_get(1,tmpX,tmpY+i)
    i=i+1
```

```
  endwhile
  ~ 横のチェック ~
  ………
  ~ 左斜めのチェック ~
  ………
  ~ 右斜めのチェック ~
  ………
next
```

で行うことができます。for~nextのループでdoubt配列からタイルを1つひとつ取り出し,それぞれの方向についてチェックするわけです。

ここまではOKでしょうか。では次に消去する方法について考えておくことにします。ディスク版YETではタイルはシュワッと色が薄くなって消えていきますね。このような消し方を実現するために,3つ以上つながっているタイルの座標を保存しておき,消すときにはこれらの座標に表示されているBGに対して一斉に処理を行うという方法を採用しました。

消せるタイルの座標は,doubt配列の未使用部分に保存しています。0~doubtEndまでは,消せるかもしれないタイルが入っていますね(これがdoubtの由来だったりする)。その後ろに付け加えるかたちで,消せるタイルの座標を蓄えていきます。縦のチェックで具体例をお見せしましょう。まず,

```
clearEnd=doubtEnd+1
```

として,消去可能なタイルの最後をセットしておきます。

そして縦のチェックの部分を,

```
i=0
while tile=bg_get(1,tmpX,tmpY+i)
  doubt(clearEnd+i,0)=tmpX
  doubt(clearEnd+i,1)=tmpY+i
  i=i+1
endwhile
if i>2 then clearEnd=clearEnd+i
```

とするわけです。タイルが何個つながろうがおかまいなしに,とにかくdoubt配列にタイルを登録していきます。そして,最後にタイルの枚数(つまりi)を調べ,3以上ならclearEndを更新するわけです。もしタイルが2枚しかつながっていなかったのならclearEndは更新されませんから,次に行われる横のチェックでは再びdoubt(clearEnd,0)から使われる

というわけです。

●checkとeraseの意味深な関係

さてこれがタイルの色を調べるアルゴリズムのすべてです。YETのプログラム中ではこの部分はcheckという名前の関数になっています。そして実際にタイルを消去するのがerase関数です。

これら2つの関数は密接に関係しあっています。まずcheck関数が色を調べ，消去できるタイルを登録します。erase関数はそれを画面から消去しますが，そのとき消去するタイルの上にあったタイルを下へ移動させます。同時にこの崩れ落ちたタイルの座標を再びdoubt配列にセットするのです。doubtEndもこのとき更新されます。そしてerase関数は，崩れ落ちたタイルがあった場合は1を，なかった場合には0を返します。したがって，

```
repeat
  check( )
  chkFlag=erase( )
until chkFlag=0
```

というループにすれば，落ちるタイルがなくなるまでチェックと消去を繰り返すという処理ができます。

チェックの高速化とビット演算

さて，インタプリタではタイルが落下したあとのチェック時間が結構かかってしまい，このままだとゲーム意欲をそがれてしまうほどです。そこでチェックルーチンの高速化を考えてみました。

上から青のタイルが3連で落ちてきた場合を考えてみましょう。check関数は一番上のタイルから調べ始めます。まずは縦方向のチェックです。ここで同じ色のタイルが3つ並んでいることがわかり，消去するためにdoubt配列に登録します。これは先ほどから説明しているとおりですね。続いて横と斜めのチェックをし，一番上のタイルに関するチェックは終了です。

次に2番目のタイルについて調べ始めます。まずは縦方向からです。ところが縦方向は一番上のタイルを調べたときにチェック済みのはずです。青の3連タイルなのですから，下向きに調べるのは無駄というものです。こういった無駄を省いてやろうというのがここでいう高速化です。タイルが崩れたあとはさまざまな場所でタイルが落下しますので，すべての方向についてチェック済みかどうかを調べるのはかなりの効果が期待できます。また実際，この処理によってかなり反応がよくなりました。

では実際にどうやってチェックを行うのかを見ていきましょう。ここでは新しい考え方であるビット演算を使います。ビット演算は2進数で考えるとよくわかります。

```
for i=0 to 10
  print bin$(i)
next
```

を実行してみてください。bin$関数は引数を2進数で表示する関数です。

2進数は0と1の2つの数字だけで成り立っています。0，1の次は，1の位をそれ以上大きくできないので繰り上がり10になるのです。このままでは位取りが見づらいので，2行目を，

```
print right$("000"+bin$(i),4)
```

に変えてみてください。今度は頭に0が補われ，桁がはっきりわかるようになったと思います。

ではビット演算を紹介しましょう。次のプログラムを入力してみてください。

```
10 end
20 /*
30 func prbin( x )
40   print right$("000"+bin$(x),4)
50 endfunc
```

これは数字を4桁の2進数で表示する関数です。関数を作りたいだけなので，10行でいきなりendになっています。これを使って実験してみましょう。

X-BASICで使えるビット演算にはnot，and，or，xorの4つがあります[1)]。

●not

notは単項演算子として使います。これは2進数の第n桁が0なら1に，1なら0にする働きがあります。

```
prbin( not &B0011 )
```

を実行してみてください。

```
1100
```

と表示されますね。

●and

これは二項演算子として使います。実例を見てみましょう。

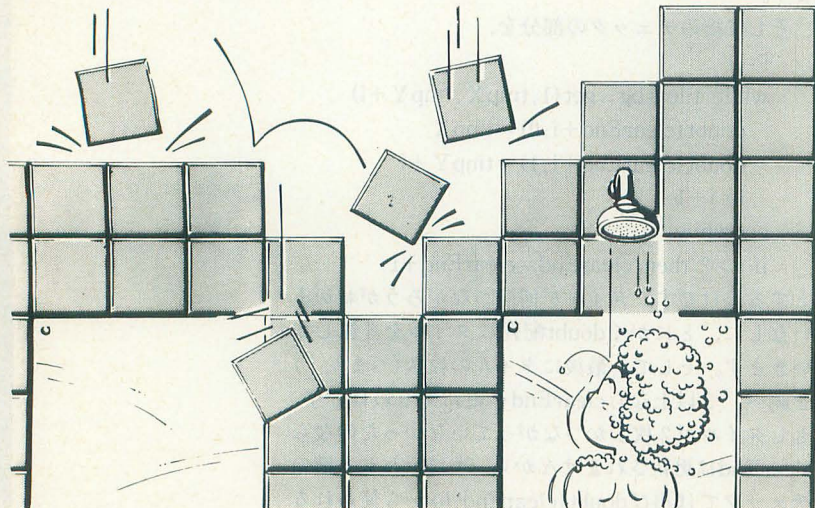
```
prbin( &B0101 and &B0011 )
```

と実行してみてください。

```
0001
```

と表示されましたね。わかりやすいように並べて見てみましょう。

```
0101
0011
────
```

1）　ビット演算にはほかにもいろいろあるのですが，実際にはnotとandさえあればどんなビット演算でも作ることができ，X-BASICはさらにorとxorまで持っているのですから困ることはないでしょう。

0001

なにか気づきませんか。そう，第n桁がどちらも1の桁だけが1になっていますね。andは目的の桁が1かどうかを調べるのに使えます。1～10の中で2桁目が1の数を知りたければ，

```
for i=1 to 10
  prbin( &B0010 and i )
next
```

とします。2桁目が1以外なら0000と表示されます。

●or

これも二項演算子として使います。実例を見てみましょう。

```
prbin( &B0101 or &B0011 )
```

を実行してみてください。

0111

と表示されます。orはどちらかの桁が1ならばその桁は1になります。そこで目的の桁を1にしたいときに使えばいいということになります。逆にandは目的の桁を0にしたいときにも使うことができます。

●xor

今回は使ってないのですが，ついでに説明しておきましょう。

```
prbin( &B0101 xor &B0011 )
```

を実行してみてください。

0110

と表示されますね。xorはどちらか一方の桁だけが1のときだけその桁が1になるのです。何に使うのか疑問に思われるかもしれませんね。いま表示された答え0110を使って，

```
prbin( &B0101 xor &B0110 )
```

としてみてください。どうですか？　ついでに，

```
prbin( &B0011 xor &B0110 )
```

も試してみましょう。

●高速化の実際

check関数ではandとorを使って，いま見ているタイルがすでにチェックされているかどうかを調べています。リストを見てください。2680行でBGのテキストページ0を0で埋めつくしています。タイルを表示しているのはテキストページ1ですから，未使用のページ0をすでにチェックされているかどうかを調べるのに使うことにしました。座標x，yに表示されているタイルのチェックが終わったら，テキストページ0の座標x，yに印を付けるわけです。

この印は各方向のチェックの最初で調べます。2780行を見てください。tmpX，tmpYというのは，これから調べようとしているタイルの座標です。テキストページ0の該当座標からデータを取り出し，1とandをとっていますね。1というのは&B0001で1桁目が1です。これは縦のチェックが終わったという印に使っています。もし1桁目が0なら，andの結果は0になりますから条件が成立し，縦のチェックが始まるというわけです。

続いて2830行を見てください。ここはwhileで同じ色が続く限り下向きにタイルを調べ続ける部分です。この2830行で調べ終わったタイルに印を付けています。orを使ってべつの桁を壊さないように印を付けている点に注意してください。

続く横のチェックでは印として2(&B0010)を，斜めのチェックでは4(&B0100)と8(&B1000)を使っています。

タイルの消去と高速化

高速化という点については，タイルのチェックよりこの消去のほうが実行価値があります。なんせこちらは消したあとのタイルの崩れという面倒な処理が待っていますからね。

check関数の実行が終わると，画面から消去すべきタイルはdoubtEnd～clearEndの間に入っていることになります。まずはこれらのタイルを画面から消さなければなりません。ディスク版YETではシュワッと消してみましたが，インタプリタでは速度の問題があってあれほど気持ちよくは消えてくれないのです。そこで白い星を表示し，「消えた！」というイメージを出してみました。リストでいうと3640行です。次の3650行が高速化を行っている部分です。

erase関数の頭，3550行を見てください。colsという配列が宣言してありますね。これは消えたタイルのx座標を保持しておくための配列です。この配列を調べれば，タイルが崩れたために残りのタイルを落下させなければならない場所がわかるというわけです。すべてのタイルを「これは落下するのかな？」などと調べていったのでは時間がかかってしかたありません（実は初期バージョンは馬鹿正直にそうやっていたのですが……）。ここでは消去したタイルのx座標に対応するcols配列に1を入れ，消去の印としています。

このあとに続く崩壊を処理しているのが3860行からのfor～nextループです。ここではタイルが消えた列（これはcols配列を調べればすぐにわかる）を下から上向きに調べ，星印と空白を飛ばしてタイルを下へ詰めていく処理を行っています。3950行のif文のorの前後が星印と空白のチェックをしている部分です。崩れ落ちたタイルはdoubt配列に登録され，再びcheckループへと戻っていくことになります。

速くなる，速くなる

ゲームが進んでいくと速度がだんだん速くなっていくというのはこのテのゲームの常です。これを仕切っているのが60行で宣言されているlevels配列と，現在のレベルを保持している変数levelです。levels配列にセットしてあるのはレベルに対応した点数で，点数が300点より大きくなればレベルは0から1へとアップします。ここはディスク版と異なっている点です[2]。

速度を上げる方法については1420行のinitFM関数を見てください。先月のものに手を加えてあり，

2)　ディスク版では出現したタイルの総数によってレベルが上がっていくようになっています。つまりディスクではタイルがあまり落ちないうちに得点をあげておくのが高得点への道となるわけです。

30本のトラックを確保するようになっています。このうち1～20のトラックに，@L192R，@L182R，……と次第に短くなっていく休符がセットしてあるのです。先月休符を演奏して待ち時間を稼ぐという方法を紹介しましたね。レベルに応じて鳴らす休符を次第に短くしてやれば，待ち時間は次第に短くなりタイルはどんどん速く落ちるようになるわけです。ポイントに応じてFM音源のチャンネル1に休符をセットしているのがerase関数の中の3770行にあるif文です。

リストの入力方法と改造

リストは先月のものを基本にしています。足りない部分を補い，check関数とerase関数を補った形になっています。補った部分は「/＊＊」で始まるコメントがついています。これらの行を追加し，2つの関数を入力すればOKです。先月号のリストを入力してある方は，今月号のリストと見比べながら補っていってください。

今月はリストの詳しい説明は行いませんでした。これは実際のリストを追いながらではアルゴリズムを十分説明できないと判断したからです。まずはアルゴリズムを頭に置いて，それがどのようなプログラムになっているのか追いかけてみてください。

levels配列は少々やさしく設定してあります。慣れてきたらスピードが上がっていくのが遅く感じることでしょう。最初の方は100点ずつレベルが上がっていく程度でいいかもしれません。YETで高得点を取るためにも，自分でlevels配列に手を入れてスピードを調整してください。

さて来月は「ファイル処理」をやりたいと思います。用意したプログラムはきっと皆さんの気に入っていただけると思います。ではまた。

リスト1

```
  10 char doubt( 200, 1 )       /** 消せる可能性のあるタイル
  20 int  doubtEnd                /** 疑ってみる最後のタイル
  30 int  clearEnd                    /** 消せる最後のタイル
  40 int  chkFlag               /** まだ消せるかを表すフラグ
  50 int  points                                    /** スコア
  60 int  levels( 19 ) = {                          /** レベル
  70   +300,   600,   900,  1200,  1500,
  80   +1800,  2100,  2400,  2700,  3000,
  90   +3500,  4000,  4500,  5000,  5500,
 100   +6000,  8000, 10000, 15000, 20000
 110 }
 120 int  level=0                        /** レベル保持用変数
 130 /*
 140 int tileX, tileY                /* タイルの座標
 150 int tmp                         /* タイル待避用
 160 int tiles( 2 )                  /* 落下中のタイル
 170 int keySen                      /* キー入力フラグ
 180 int gameOver
 190 int i
 200 str ch
 210 /*
 220 screen 1,3,1,1                  /* 512×512×65536色
 230 sp_init()                       /* スプライトの初期化
 240 sprite_pattern()                /* パターンの設定
 250 sprite_color()                  /* パレットの変更
 260 sp_disp( 1 )                    /* スプライト表示
 270 bg_set( 0, 1, 1 )               /* ページ1を表示
 280 initFM()                        /* FM音源の設定
 290 makeScreen()                    /* 盤面作り
 300 /*
 310 locate 22,29                    /** 乱数を
 320 print "キーを押してください"      /** 初期化する
 330 while inkey$(0)=""              /** ループ
 340   rnd()                         /**
 350 endwhile                        /**
 360 locate 22,29                    /** メッセージを
 370 print chr$(5)                   /** 消す
 380 while 1
 390   tileX = 16                    /* 最初の位置
 400   tileY = 3
 410   keySen = 1                    /* キー入力可
 420   /*
 430   gameOver = 0                  /* ゲーム続行
 440   for i=0 to 2
 450     if bg_get( 1, tileX, tileY+i ) <> 256 then {
 460       gameOver = 1  /* 表示位置にタイルがあれば終わり
 470       break
 480     }
 490                                 /* 新しいタイル生成
 500     tiles( i ) = int( rnd()*6 + 2 )*256 + 1
 510     bg_put( 1, tileX, tileY+i, tiles(i) )
 520   next
 530   if gameOver then break        /* whileを抜ける
 540   /*
 550   m_play( 1 )                   /* ウェイト開始
 560   while inkey$(0)<>"" : endwhile /* 先行入力をクリア
 570   while 1
 580     if ( keySen ) then ch = inkey$(0) /* キー入力
 590     switch ( asc(ch) )
 600       case '4'                  /* 右へ移動
 610         moveTile( tileX-1 ) : break
 620       case '6'                  /* 左へ移動
 630         moveTile( tileX+1 ) : break
 640       case '5'                  /* タイル回転
 650         tmp = tiles( 2 )
 660         bg_put( 1, tileX, tileY+2, tiles( 1 ))
 670         tiles( 2 ) = tiles( 1 )
 680         bg_put( 1, tileX, tileY+1, tiles( 0 ))
 690         tiles( 1 ) = tiles( 0 )
 700         bg_put( 1, tileX, tileY, tmp )
 710         tiles( 0 ) = tmp
 720         break
 730       case ' '                  /* タイルを落とす
 740       case '2'
 750         keySen = 0
 760         break
 770     endswitch
 780     if ch = chr$(27) then {      /* 一時停止
 790       while inkey$(0)<>"" : endwhile
 800       while inkey$(0)="" : endwhile
 810     }
 820     /*
 830     /* ウェイト終了、またはタイル落下の処理
 840     /*
 850     if m_stat( 1 )=0 or keySen=0 then {
 860       if bg_get( 1, tileX, tileY+3 ) = 256 then {
 870         for i=0 to 2  /* 下にタイルがなければ1つ下へ
 880           bg_put( 1, tileX, tileY+i, 256 )
 890         next
 900         tileY = tileY + 1
 910         for i=0 to 2
 920           bg_put( 1, tileX, tileY+i, tiles(i) )
 930         next
 940       }
 950       m_play( 1 )                 /* 再びウェイトをかける
 960     }
 970     if bg_get( 1, tileX, tileY+3 ) <> 256 then break
 980   endwhile
 990   /* if gameOver then break
1000   for i=0 to 2  /** 3つのタイルをcheck配列に登録する
1010     doubt( i, 0 ) = tileX               /** X座標を保存
1020     doubt( i, 1 ) = tileY+i             /** Y座標を保存
1030   next                                  /**
1040   doubtEnd = 2                     /** 疑ってみるのは0～2
1050   repeat                                /**
1060     check()              /** 消せるかどうかチェックし
1070     chkFlag = erase()                        /** 消去する
1080   until chkFlag = 0     /** 消せるものがなくなるまで
1090 endwhile
1100 end
1110 /*
1120 func moveTile( newx )              /* タイルの横への移動
1130   int i
1140   for i=0 to 2                      /* 横に障害物はないか
1150     if bg_get( 1, newx, tileY+i ) <> 256 then return()
1160   next
1170   for i=0 to 2                      /* なければ移動
1180     bg_put( 1, tileX, tileY+i, 256 )
1190   next
1200   for i=0 to 2
1210     bg_put( 1, newx, tileY+i, tiles(i) )
1220   next
1230   tileX = newx                      /* tileXを更新
1240 endfunc
1250 /*
1260 func makeScreen()
1270   int i
1280   fill( 0, 0, 511, 511, rgb( 0, 16, 0 ))
1290   bg_fill( 1, 256 )                 /* BGを256で埋める
1300   for i=5 to 26                     /* レンガを縦に積む
1310     bg_put( 1, 11, i, 256*8+2 )
1320     bg_put( 1, 21, i, 256*8+2 )
1330   next
1340   for i=11 to 21                    /* レンガの床を作る
1350     bg_put( 1, i, 26, 256*8+2 )
1360   next
1370   color 7                           /* 白の太文字
1380   locate 25, 1
1390   print using "Points  ######";0 /* 消したタイル数
1400 endfunc
1410 /*
1420 func initFM()
1430   int i                                   /** 変数追加
1440   for i=1 to 30
1450     m_alloc( i, 40 )            /** トラックを30本確保
1460   next
1470   for i=1 to 20           /** 20本のトラックに
1480     m_trk( i, "@L"+str$((20-i)*10+2)+"R" )
1490   next                    /** 次第に速くなる休符をセット
1500   /*
1510   m_assign( 1, 1 )
1520   m_tempo( 200 )
1530   m_trk( 25, "V15@5L16CCC" ) /** ポイント時のベルの音
1540   m_trk( 26, "V15@5L16EFE" ) /** ポイント時のベルの音
1550   m_trk( 27, "V15@5L16GAG" ) /** ポイント時のベルの音
1560   m_assign( 2, 25 )             /** ベルの音をセット
1570   m_assign( 3, 26 )             /** ベルの音をセット
1580   m_assign( 4, 27 )             /** ベルの音をセット
1590 endfunc
1600 /*
1610 func sprite_color()
1620   int i, j
1630   int r, g, b
1640   int cr, cg, cb                    /* Color of Red … …
```

```
1650   int  colors(17) = {              /** 最も暗い色の配列
1660      +0,  0, 15,  /** blue
1670      +0, 10,  0,  /** green
1680      +0, 15, 15,  /** cyan
1690     +15,  0,  0,  /** red
1700      +8,  0, 15,  /** magenta
1710     +15, 15,  0   /** yellow
1720   }
1730   int colorBase = 0  /** colors配列中の該当位置を保持
1740   /*
1750   for i=1 to 6                  /* 2進数でいうと010 ～ 110
1760     r = i / 4                        /* 1・・ を判定
1770     g = (i mod 4) / 2                /* ・1・ を判定
1780     b = i mod 2                      /* ・・1 を判定
1790     for j=0 to 4
1800       cr = r * (j*4 + colors(colorBase))     /** 変更
1810       cg = g * (j*4 + colors(colorBase+1))   /** 変更
1820       cb = b * (j*4 + colors(colorBase+2))   /** 変更
1830       sp_color( j+1, rgb( cr, cg, cb ), i+1 )
1840     next
1850     colorBase = colorBase + 3
1860     sp_color( 6, 1, i+1 )              /* 6は黒
1870   next
1880                                      /* 枠の色作り
1890   for i=0 to 2
1900     sp_color( i+1, hsv(16,31,i*2+12 ), 8 )
1910   next
1920   sp_color( 4, hsv(22,31,22), 8 )
1930                                       /** 消去パターンの色
1940   for j=0 to 4
1950     sp_color( j+1, rgb( j*4+15, j*4+15, j*4+15 ), 9 )
1960   next
1970 endfunc
1980 /*
1990 func sprite_pattern()
2000   dim char c(255)
2010   int i
2020   c={
2030     +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
2040     +15, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,14,
2050     +15, 6, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 6,14,
2060     +15, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6,14,
2070     +15, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6,14,
2080     +15, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6,14,
2090     +15, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6,14,
2100     +15, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6,14,
2110     +15, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6,14,
2120     +15, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6,14,
2130     +15, 6, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6,14,
2140     +15, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6,14,
2150     +15, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6,14,
2160     +15, 6, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 6,14,
2170     +15, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,14,
2180     +14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14
2190   }
2200   sp_def(1,c)
2210   c = {
2220     +3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
2230     +3, 4, 4, 4, 4, 4, 4, 1, 3, 4, 4, 4, 4, 4, 4, 1,
2240     +3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1,
2250     +3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1,
2260     +3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1,
2270     +3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1,
2280     +3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1,
2290     +3, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1,
2300     +3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
2310     +4, 4, 4, 1, 3, 4, 4, 4, 4, 4, 4, 1, 3, 4, 4, 4,
2320     +2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2,
2330     +2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2,
2340     +2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2,
2350     +2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2,
2360     +2, 2, 2, 1, 3, 4, 2, 2, 2, 2, 2, 1, 3, 4, 2, 2,
2370     +1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1
2380   }
2390   sp_def(2,c)
2400   c={                                   /** 消去用パターン追加
2410     +0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
2420     +0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0,
2430     +0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0,
2440     +0, 0, 0, 5, 0, 0, 3, 4, 3, 0, 0, 5, 0, 0, 0, 0,
2450     +0, 0, 0, 0, 5, 2, 3, 4, 3, 2, 5, 0, 0, 0, 0, 0,
2460     +0, 0, 0, 0, 2, 3, 4, 5, 4, 3, 2, 0, 0, 0, 0, 0,
2470     +0, 0, 0, 3, 3, 4, 4, 5, 4, 4, 3, 3, 0, 0, 0, 0,
2480     +0, 3, 3, 4, 4, 5, 5, 5, 5, 5, 4, 4, 3, 3, 0, 0,
2490     +0, 0, 0, 3, 3, 4, 4, 5, 4, 4, 3, 3, 0, 0, 0, 0,
2500     +0, 0, 0, 0, 2, 3, 4, 5, 4, 3, 2, 0, 0, 0, 0, 0,
2510     +0, 0, 0, 0, 5, 2, 3, 4, 3, 2, 5, 0, 0, 0, 0, 0,
2520     +0, 0, 0, 5, 0, 0, 3, 4, 3, 0, 0, 5, 0, 0, 0, 0,
2530     +0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0,
2540     +0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0,
2550     +0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
2560     +0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
2570   }
2580   sp_def(3 ,c)
2590 endfunc
2600 /*
2610 func check()
2620   int tmpX, tmpY
2630   int tileColor
2640   int tile, i, j, counter
2650   int tX, tY      /* 起点移動用
2660   /*
2670   clearEnd = doubtEnd + 1
2680   bg_fill( 0, 0 )      /* 調べた方向を保持するフラグ用
2690   for tile=0 to doubtEnd
2700     tmpX = doubt( tile, 0 )
2710     tmpY = doubt( tile, 1 )
2720     doubt( tile, 0 ) = 0
2730     doubt( tile, 1 ) = 0
2740     tileColor = bg_get( 1, tmpX, tmpY )
2750     /*
2760     /* 縦のチェック
2770     /*
2780     if ( bg_get( 0, tmpX, tmpY ) and 1 ) = 0 then {
2790       i = 0
2800       while tileColor = bg_get( 1, tmpX, tmpY+i )
2810         doubt( clearEnd+i, 0 ) = tmpX
2820         doubt( clearEnd+i, 1 ) = tmpY+i
2830         bg_put(0,tmpX,tmpY+i,bg_get(0,tmpX,tmpY+i) or 1
2840         i = i+1
2850       endwhile
2860       if i > 2 then clearEnd = clearEnd+i
2870     }
2880     /*
2890     /* 横のチェック
2900     /*
2910     if ( bg_get( 0, tmpX, tmpY ) and 2 ) = 0 then {
2920       tX = tmpX : tY = tmpY
2930       while tileColor = bg_get( 1, tX, tY )
2940         tX = tX - 1
2950       endwhile
2960       tX = tX + 1
2970       i = 0
2980       while tileColor = bg_get( 1, tX+i, tY )
2990         doubt( clearEnd+i, 0 ) = tX+i
3000         doubt( clearEnd+i, 1 ) = tY
3010         bg_put( 0, tX+i, tY, bg_get( 0, tX+i, tY ) or 2 )
3020         i = i+1
3030       endwhile
3040       if i > 2 then clearEnd = clearEnd+i
3050     }
3060     /*
3070     /* 左斜め上のチェック
3080     /*
3090     if ( bg_get( 0, tmpX, tmpY ) and 4 ) = 0 then {
3100       tX = tmpX : tY = tmpY
3110       while tileColor = bg_get( 1, tX, tY )
3120         tX = tX - 1
3130         tY = tY - 1
3140       endwhile
3150       tX = tX + 1
3160       tY = tY + 1
3170       i = 0
3180       while tileColor = bg_get( 1, tX+i, tY+i )
3190         doubt( clearEnd+i, 0 ) = tX+i
3200         doubt( clearEnd+i, 1 ) = tY+i
3210         bg_put( 0, tX+i, tY+i, bg_get( 0, tX+i, tY+i ) or 4 )
3220         i = i+1
3230       endwhile
3240       if i > 2 then clearEnd = clearEnd+i
3250     }
3260     /*
3270     /* 右斜め上のチェック
3280     /*
3290     if ( bg_get( 0, tmpX, tmpY ) and 8 ) = 0 then {
3300       tX = tmpX : tY = tmpY
3310       while tileColor = bg_get( 1, tX, tY )
3320         tX = tX + 1
3330         tY = tY - 1
3340       endwhile
3350       tX = tX - 1
3360       tY = tY + 1
3370       i = 0
3380       while tileColor = bg_get( 1, tX-i, tY+i )
3390         doubt( clearEnd+i, 0 ) = tX-i
3400         doubt( clearEnd+i, 1 ) = tY+i
3410         bg_put( 0, tX+i, tY+i, bg_get( 0, tX-i, tY+i ) or 8 )
3420         i = i+1
3430       endwhile
3440       if i > 2 then clearEnd = clearEnd+i
3450     }
3460   next
3470   clearEnd = clearEnd - 1
3480 endfunc
3490 /*
3500 func int erase()
3510   int retVal = 0
3520   int i, j, btm /* bottom
3530   int tile
3540   int palColor, newColor
3550   int cols( 8 ) = { 0, 0, 0, 0, 0, 0, 0, 0, 0 }
3560   int counter = 0
3570   /*
3580   if clearEnd = doubtEnd then return( 0 )
3590   /* m_play( 2, 3, 4 )
3600   /*
3610   /* clear tile
3620   /*
3630   for i=doubtEnd+1 to clearEnd
3640     bg_put( 1, doubt(i,0), doubt(i,1), 256*9+3 )
3650     cols( doubt(i,0)-12 ) = 1
3660     counter = counter + 1
3670   next
3680   /*
3690   if counter > 0 then {
3700     if counter = 3 then {
3710       points = points + 65
3720     } else {
3730       points = points + counter*20
3740     }
3750     locate 33, 1
3760     print using "######";points
3770     if points > levels( level ) then {
3780       if level < 19 then level = level+1
3790       m_assign( 1, level+1 )
3800     }
3810   }
3820   repeat : until m_stat(2) = 0
3830   /*
3840   doubtEnd = -1
3850   m_play( 2, 3, 4 )
3860   for i=12 to 20
3870     if cols( i-12 ) = 0 then continue
3880     j = 25
3890     while bg_get( 1, i, j ) <> 256*9 + 3
3900       j = j - 1
3910     endwhile
3920     btm = j
3930     while j > 2
3940       tile = bg_get( 1, i, j )
3950       if tile = 256 or tile = 256*9+3 then {
3960         if tile = 256*9+3 then bg_put( 1, i, j, 256 )
3970         j = j - 1
3980         continue
3990       }
4000       bg_put( 1, i, btm, tile )
4010       doubtEnd = doubtEnd + 1
4020       doubt( doubtEnd, 0 ) = i
4030       doubt( doubtEnd, 1 ) = btm
4040       bg_put( 1, i, j, 256 )
4050       btm = btm - 1
4060       j = j - 1
4070     endwhile
4080   next
4090   return( 1 )
4100 endfunc
```

PurePASCAL

PASCALプログラミングへの招待〈2〉

PASCALの特徴的な性格について

Fujiki Takeshi　Fujii Yoshimi
藤木健士/藤井義巳

PurePASCALは6月号で発表されたX68000用のPASCALコンパイラです。PASCAL言語の特徴的な概念について解説します。PurePASCALの使い方は，6月号の付録ディスクのPASCAL.MANというファイルを参照してください。

先月までのところでPASCALのだいたいの感じがわかったと思いますので今月はもう少しつっこんだ話をすることにしましょう。PASCALの特徴について述べてみたいと思います。FORTRAN全盛の時代に生まれたPASCALはそれまでになかった多くの考え方を導入しました。これによりプログラマは以前よりも安全性の高いプログラムを書けるようになり，以前は記述が困難であったアルゴリズムを記述できるようになったのです。それを可能にしたのは入れ子構造を反映したスコープ(有効範囲)，CALL BY VALUE，再帰呼び出しといったPASCALの持つ特徴です。これらのものは比較的新しいプログラミング言語においては常識となっているといっても過言ではありません。それでは今回はこれらの特徴に関して説明していきたいと思います。

スコープ

スコープとは定数，変数，関数および手続きの定義あるいは宣言された名前の見え方のことです。簡単な例を示してみましょう

```
 1:  PROGRAM TEST(INPUT, OUTPUT);
 2:
 3:  VAR  I,F:INTEGER;
 4:
 5:  PROCEDURE FACT;
 6:     VAR J: INTEGER;
 7:
 8:       BEGIN
 9:          F: =1;
10:          FOR J:=1 TO I DO
11:               F:=F*J
12:       END;
13:
14:  BEGIN
15:       WRITELN('INPUT I');
16:       READ(I);
17:       FACT;
18:       WRITELN('I=',I,' FACT=', F)
19:  END.
```

(注意：行番号は説明のためにつけてありますが実際のプログラムにはありません)

簡単にプログラムの説明をしましょう。このプログラムは階乗を求めるプログラムです。このプログラムでは12行目までが宣言部で14行以降が本体になっています。それでこのプログラムは次のように動作します。

1)　'INPUT I' を表示する
2)　Iをキーボードから読み込む
3)　階乗を計算する手続きをコールする
4)　結果を表示する

したがって，ステートメントの実行の順序は行番号でいうと15，16，9，10，11，18という順になります。FACTという手続きはメインルーチンよりも上にありますが，これは宣言されているだけであって実際に実行されるのはそれが呼び出されたときです。それで実行の順序は上のようになるのです。

このプログラムのようにPASCALでは定数，型，変数，手続きおよび関数は使用する前にすべて宣言されていなければなりません。それではWRITE, READ, INTEGERなどはどうなのでしょうか。これらは事前に定義あるいは宣言されたものとして処理系が準備しているので宣言なしに使用してよいのです。

話がそれてしまいましたがスコープルールに話を戻しましょう。上でスコープルールとは名前の見え方に関する規則だといいましたが，次のように定義することができます。

　名前はそれが定義または宣言されたブロックの中で有効である。

ブロックとは宣言部と本体から構成され，プログラム，手続きおよび関数はひとつのブロックから構成されます。上のプログラムの場合，1行目から19行目までがプログラムブロックで，5行目から12行目までが手続きFACTのブロックです。ということはこの手続きFACTは当然プログラムブロックに含まれていることになります。

それではこのプログラムの中で扱われている変数を例にとって具体的に説明することにしましょう。このプログラムの中では変数はI，Fがプログラムブロックで宣言されJがFACTという手続きのブロック中で宣言されています。IとFはプログラムブロックで有効ですから，プログラム中のどこからでも見えます（アクセスできる状態のことを“見える”とよくいいます)。それで手続きの中であっても9行目や11行目で変数Fにアクセスできるわけです。

このようにプログラムブロックで宣言された変数はプ

ログラム中のどこからでも見えますから，このことをグローバル変数（大域的な変数）と呼ぶこともあります。それに対してJは手続きの中で宣言されていますからこのブロックの中でしか使うことができません。

試しに14から19行目のあいだにJ：＝1；というステートメントを入れてコンパイルしてみるとJが宣言されていないとエラーが出るでしょう。それでは次のような場合はどうなるのでしょうか。

```
 1: PROGRAM TEST(INPUT, OUTPUT);
 2:
 3: VAR        I, F: INTEGER;
 4:
 5: FUNCTION FACT(X: INTEGER):INTEGER;
 6:      VAR I,J:INTEGER;
 7:
 8:      BEGIN
 9:           I:=1;
10:           FOR J:=1 TO X DO
11:                I:=I*J;
12:           FACT:=I
13:      END;
14:
15: BEGIN
16:      WRITELN('INPUT X');
17:      READ(I);
18:      F:=FACT(I);
19:      WRITELN('I=', I,' FACT=', F)
20: END.
```

この例の中でIという変数が2回宣言されています(3行目と6行目)。さてこのプログラムはどのように動作するのでしょうか。この場合PASCALは内側のブロックで宣言された変数を有効にします。それで上のプログラムでは手続きFACTのブロックではプログラムブロックで宣言された変数Iは有効でなくFACTの中で宣言された変数が有効となります。つまり同じ名前が定義あるいは宣言されたときはその名前を使うブロックから見て一番内側のブロックで宣言されたものが見えるのです。6行目の変数Iの宣言をしないと，19行目で結果を書き出すときIの値がおかしくなるでしょう。それはFACTの中でプログラムブロックで宣言したIを更新してしまうからです。

この程度の短いプログラムだとなにがおかしいか探すのは容易ですが，長いプログラムだとそうもいきません。皆さんもこれに似た経験をしたことがありませんか？変わるはずのない変数の値が知らないうちに変わっていてうまくプログラムが動かない！　調べてみると実は別のところで使っている変数名を使ってしまっていたということはありませんか？　この手のバグは探すのが非常に困難です。

でもスコープがあるために，このような心配をする必要がなくなります。変数を使用するブロックで変数を宣言してしまえば，上位のブロックで同じ名前の変数があったとしてもそれが更新される心配をしなくていいということです。とにかく変数の見える範囲を選択できるのはプログラマにとって非常に有利な条件です。“変数はプログラムブロックで宣言すればいいや”なんて考えないでスコープを考慮して安全はプログラムを書くようにしましょう。ここでは変数を例にとって説明しましたが，定数，型，手続きおよび関数に関してもスコープは適用されます。

CALL BY VALUE

今度はCALL BY VALUEという考え方について説明しましょう。これは手続きや関数を呼び出すときに渡す引数に関する規則です。ここでも例を挙げて説明することにします。

```
 1: PROGRAM TEST(INPUT, OUTPUT);
 2:
 3: VAR        I,F:INTEGER;
 4:
 5: FUNCTION FACT(X:INTEGER):INTEGER;
 6:
 7:    BEGIN
 8:        FACT:=1;
 9:        WHILE X > 1 DO
10:          BEGIN
11:            FACT:=FACT*X;
12:            X:=X-1
13:          END
14:      END;
15:
16: BEGIN
17:      WRITELN('INPUT I');
18:      READ(I);
19:      F:=FACT(I);
20:      WRITELN('I =',I,' FACT=', F)
21: END.
```

これをFORTRANで書くと次のようになります。

```
 1: PROGRAM TEST
 2: INTEGER I, F
 3:
 4: WRITE(6, *)'INPUT I'
 5: READ(5, *) I
 6: F=FACT(I)
 7: WRITE(6, *)'I=', I,' FACT=',F
 8: STOP
 9:
10: INTEGER FUNCTION FACT(X)
11: INTEGER X
12:
13: FACT=1
14: 10 IF(X.GT.1) RETURN
15:    FACT=FACT*X
16:    X=X-1
```

```
17:      GOTO 10
18:
19:      END.
```

さてこれらはまったく同じ結果を出すような感じですが実際はそうではありません。結果からいうと，FORTRANで書かれたほうのプログラムでは7行目の表示の部分で必ず'I=1　FACT=XXXX'と表示してしまいます。Iが必ず1になるのです。さてこれはどうしてでしょうか？　PASCALとFORTRANの違いはどこにあるのでしょうか。

それは，FORTRANは変数のアドレスを渡し，PASCALは値を渡すという点です。上のPASCALの場合19行目で関数FACTに実引数としてIを渡していますが，これはIの値をFACTの仮引数Xにコピーして関数をコールしていると考えるとよいでしょう。それでPASCALのプログラムでは手続きの中でXの値を更新しても（12行目）Iの値は変わらないのです。

それに対してFORTRANのプログラムでは引数として変数のアドレスを渡すので，6行目のように実引数をIとしてFACTをコールした場合，Iのアドレスが引き渡されます。そして呼び出された関数FACTの中では，Xにアクセスするステートメントでは引数として与えられた変数のアドレスにアクセスします。例のプログラムでは6行目でFACT(I)というかたちで関数FACTの引数としてIを渡しているので，FACT内の14行目15行目では変数Iを参照し，16行目では変数Iから1を引くのです。つまり変数Xの部分がすべてIに置き換えて処理をしたのと同じ結果を得るのです。それで16行目で1引くという処理は変数Iに対して行われ，結果としてI＝1となってしまいます。

PASCALのように引数を値で渡すことをCALL BY VALUEと呼び，FORTRANのように引数をアドレスで渡すことをCALL BY REFERENCEと呼びます。CALL BY VALUEでは，関数や手続きを呼び出すときにパラメータとして渡した変数が更新されることがなくなります。これにより，より安全なプログラムを書くことができます。

ちなみにPASCALでもパラメータを値としてではなくアドレスとして渡すことができないわけではありません。たとえば上のPASCALのプログラムでパラメータをアドレスとして渡すようにしたいときは次のように書きます。

```
5: FUNCTION FACT(VAR X:INTEGER):INTEGER;
```

こうするとFORTRANのプログラムと同様の結果が得られるでしょう。複数の情報を呼び出した側に返すような関数を書きたいときなどにはこれを使うとよいでしょう。

再帰呼び出し(RECURSIVE CALL)

手続きや関数が自分自身を呼び出すことができるとアルゴリズムが記述しやすい場合があります。たとえばQUICK SORTなどはループで記述するよりも簡単に書くことができるのではないでしょうか？　このように自分自身をコールすることを再帰呼び出しといいます。先の階乗を求めるプログラムを再帰呼び出しを用いて下のように書くことができます。

```
 1:  PROGRAM TEST(INPUT, OUTPUT);
 2:
 3:  VAR     I, F:INTEGER;
 4:
 5: FUNCTION FACT(I:INTEGER):INTEGER;
 7:      BEGIN
 8:          IF I > 1 THEN
 9:               FACT:=FACT(I-1)*I
10:          ELSE
11:               FACT:=1
12:      END;
13:
14:  BEGIN
15:      WRITELN('INPUT I');
16:      READ(I);
17:      F:=FACT(I);
18:      WRITELN('I=',I,' FACT=', F)
19:  END.
```

この9行目で関数FACTの中からFACTを呼び出しています。これは，

N>1 のとき

N!=N*(N−1)!

N=1 のとき

N!=1

であることを記述しているのです。このように，数学の漸化式などの値を求めたいときには再帰呼び出しは便利です。最初は再帰のプログラムは難しいと感じますが，実際にこれを使ったほうが書きやすい場合がしばしばあります。読者の皆さんも使わず嫌いにならずに積極的にこれを使ってみることをおすすめします。再帰のプログラムは終了条件さえちゃんと書ければうまく動くものです。

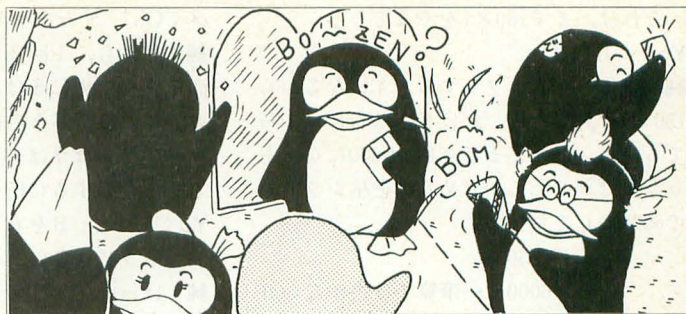
この再帰呼び出しもFORTRANにない機能です。CALL BY VALUEと深い関係があり，それが実現されることによって簡単に再帰が可能になったのです。

最後に

今回はPASCALという言語の持つ特徴の一部をそれ以前の言語と比較して紹介しました。今回説明した部分は私たちがPurePASCALを作るときにもけっこう悩んだ部分です。上で挙げた程度のものならいいのですが，再帰したときのスコープはどうなるかと，2人で激論を戦わせたこともありました。結局わからないので大学の先生に電子メールを出して教えてもらったこともあります。これらに関しては非常に奥の深いものがあります。読者の皆さんもこのあたりのことに関心があるなら専門書を読んでみられてはいかがでしょうか。きっと得るものも多いと思います。

マシン語カクテル in Z80's Bar

第13回──電卓はビットシフトで──

シナリオ＆イラスト：山田純二
特別監修：浦川博之　金子俊一

なんだかんだで，めでたく１周年を迎えてしまったZ80's Bar。１周年ということは13回目ということである。……不吉だ。しかしここに集まった面々にはそんなことはまったく関係なく，いつもどおり脳天気にお話を進めていくのであった。

愛されて１周年

カランコロ～ン

純二（以下純）：こんばんわ～。おっ，みんな集まってますね。

古村（以下で）：あっ，不良の山田君。

純：（ドテッ）な，なんですかそれは……。

善司（以下善）：念願かなって買ったX68000が初期不良だったんでしょ。

純：い，いつの間に知れわたったんだ……。

マスター（以下M）：酒のサカナに私がいいふらしたんですよ，はっはっは。

純：ひどいなぁ，せっかくツケを払いにきてあげたっていうのに。

長老（以下老）：まぁまぁ，そんな話はひとまず置いといて，こっちでパーッとやらんかの？

純：あれ，どうしたんです長老。手にメガホンなんか持っちゃって。

老：こんなハデなメガホンがあるか。パーティだから盛り上げようとしとるんじゃないか。

純：へっ？　なんのパーティ？

ようこ（以下Yo）：あら，純二君気づいてないの？

で：もちろんショートプロぱー……。

源光（以下光）：KATINOっ！

で：ぐー……。

老：実はじゃな，この７月号をもって，Z80's Barは，

純：最終回！

老：１周年じゃ，ばかもの！　いきなり縁起の悪いことをいうでない，まったく。（コホン）そういうわけで読者のみなさん，これからもZ80's Barを，

一同：よろしくおねがいしまーす。

善：いえいえこちらこそ。

メアリー（以下メ）：Come on！　じゅんじモ，いっしょに騒グデース。

光：JAZZ'INもメッコールもあるよ。

純：……。

で：みんなー，長老のおごりだからどんどんいこー。

老：女の子の分だけだといったじゃろうが。

善：そうよ，男の子はだめよ，ねぇ長老。

老：南氷洋に沈めてほしいのか？

純：ねぇ，なんか１周年のプレゼントとか記念品はないんですか？

M：山田君にはツケ払いに，プログラムの宿題をどっさり差しあげます。

純：（ちぇっ）へーい。一応ひとつ作ってきたんですけど……。

メ：What kind of？

純：電卓……。

で：ただの？

老：まぁそういうな。電卓にも四則演算などの重要なテクニックが入ってるんじゃぞ。復習の意味もかねて見せてもらおうかの。

清く正しい乗除算

老：まずは乗除算のルーチンじゃな。

で：これは，足し算の繰り返しと引き算の繰り返しをしてるのかな？

光：確かに基本はそのとおりだけど，ちょっと情けないですね。

純：ははは……。

老：ビットシフトを使ったアルゴリズムぐらいはやってほしいところじゃな。

純：そういえば学校の授業でも聞いたことあるような……。

M：どうせ，寝てるかマンガでも読んでたんでしょう。

純：睡眠学習といってください。馬鹿にしないでくださいよ，僕はこれで歴史を覚えたんですから。

善：鳴くよ（794）ウグイス，

純：平安京エイリアン。

光：筋金入りだね。

老：……とりあえず２進数８ビットの掛け算をやってみなさい。

純：10－10（テン－テン）（了解）。

```
        10010110   ……A
×)      00101001   ……B
        10010110
     10010110
  10010110
  1100000000110   ……答
```

となります。

老：よろしい。これを見てなにかわかるかな。

で：みんな１と０ばっかりです。

光：だーかーらぁー。

Yo：１のあるところだけ計算してるんじゃない？

老：そのとおり。

1)　Bが１のビットを探す。
2)　そのビットの桁にAを書く。
3)　その桁のまま足す。

と，こういう順番になるな。

光：ちなみにAのほうを被乗数，Bのほうは乗数という。

善：人食いザメのことか。

で：それはジョーズでしょ。

善：あら，お上手。なんちて。

光：だぁーっ，お前らはっ。

純：マシン語では，えーと，まずBを右に１ビットシフトする。キャリフラグでビットの0/1を調べて，0ならなにもしないで，１ならAを足す。それからAを左に１ビットシフトさせて，Bが０になるまで繰り返すんですね。

Yo：Aの左シフトってのはどうして？

純：10進数で見てみればわかりますよ。

```
    150
×)   41
```

これは，まず150×1をやるでしょ。
Yo：ふんふん。
純：次に41を右シフトすると4になるから150×4を計算する。だけど桁は右につまってるから，このままじゃ150＋600になっちゃう。だからあらかじめ600を左シフトして6000にしておけば，

```
  150
 6000  ←筆算では書かない0
─────
 6150
```

となるわけ。
Yo：純二君すごーい。
純：いやぁ，ははは。
光：筆算どおりにやるならそのとおりだけど，ほかのやり方もあるんですよね，長老。
老：うむ，乗数（B）と答えを左シフトして求めるやり方じゃな。
純：へー，どうやるの？
光：まず，Bと答えを左に1ビットシフトさせる。キャリフラグでビットの0/1を調べ，0のときにはなにもせずにループを繰り返し，1のときには答えにAを足してやる。これを8回繰り返せばできあがり。
純：どう違うんですか？
老：最初の方法は繰り返しが最大でも8回なのでスピードは速いが，プログラムが複雑なのじゃ。後者の方法は必ず8回繰り返すから若干遅いがプログラムは組みやすいな。
善：サンプルは後ろの方法だね。
老：どうじゃ，足し算だけでやるよりはエレガントじゃろう。
善：確かにお鼻が長いなぁ。
で：そりゃエレファントでしょうが。

＊　＊　＊

M：次は割り算ですね。
老：こちらのほうも，まずは筆算でやってみなさい。
純：らじゃ～。

```
             11……商
        ──────────
101001) 10010110←A
  ↑      101001
  B     ───────
         1000100
          101001
         ───────
   余り……11011
```

（Aが被除数，Bが除数）
できましたよ。
老：よし。この筆算では，なにがどっちにシフトしながら計算を繰り返しているかわかるかな。
純：えーっとですね，1回計算するたんびにBがだんだん右にシフトして，商は左にシフトしてます。
Yo：なんだかよくわかんなーい。
メ：Oh！　サッキトオナジヨ。
純：だから，1回商が立つたんびに，次の商を書くために商はひと桁上げてるってこと。これはさっきの掛け算と同じ理屈。で，除数のBのほうは書く位置が右にずれてるでしょ。要するに桁を小さくしてるわけ。
Yo：だって，Bを右シフトしたら数が変わっちゃうじゃない？
純：げっ……。
老：ほっほっほ。ようこちゃんのいうとおりじゃ。割り算はいちばん左の桁から計算を始めるんじゃから，最初にBをAの1歩手前まで左シフトせんと計算が始められんな。
純：そうか，わかった。
1)　左シフトでBをスタート位置につける。
2)　商を左にシフトする。
3)　AがシフトY前のBより小さくなってたら計算終了。
4)　AからBを引いてみて，引けたら商を＋1。だめだったらAを復元する。
5)　Bを右シフトする。
6)　2へ戻る。
これなら100点でしょう？
老：うむ。実際のプログラムを見ると，このアルゴリズムとは一見違って見えるかもしれんが，基本の部分は同じじゃぞ。

変換しようよ！

純：ふー。なんとか乗除算はできた。
老：まだまだ先は長いぞ。電卓なら，打ち込まれた数字をコンピュータにわかる数値に変換せねばならん。
純：あ，それなら作ってきたこれで大丈夫だと思います。
老：フムフム……。純二君，ループの先頭でHLレジスタを10倍しているのはなぜかの？
純：左端の桁から変換してるからです。
Yo：そんなこといったってわかんなーい。
純：つまり，10進数で3624という数は，3624＝$10^3×3+10^2×6+10^1×2+10^0×4$というふうに表すことができるでしょ。それで，N桁目の数字は$10^{(N-1)}$倍されているというのを使うわけ。1回目のループではHL＝3で，次の桁に移ったときには，HLを10倍してHL＝3×10＋6になる。ループを繰り返すごとにHL＝$3×10^2+6×10+2$……最後にはHL＝3624となるんだ。でしょ，長老。
老：信ずる者は救われる，とは限らんが正解じゃ。それぞれの桁の重みを考慮することが大切なんじゃよ。
M：それじゃあ今度は，数値から数字への変換ですね。
光：つまり16進数から10進数への変換なんだけど，純二君，一般の基数変換の授業は聞いてたかい？
純：知ってる知ってる，これだけはさんざんやらされたんで，覚えてますよ。
光：よし，じゃあ説明してもらおう。
純：基本は割り算の繰り返しです。変換したい数値を，基数で割っていって，余りで下の桁から埋めていくんですよね。
老：なぜ下からか説明できるかな？
純：割り算をするたびに商は1桁上のかたまりになるからかな？　3624割る10だと4が余って，362を10で割ると2が余って……という10進数→10進数の変換で見ればなんとなくわかると思いますけど。
老：具体的には？
純：3624（10進数）を16進数に変換するには，

```
16) 3624     余り
16)  226 ……8
      14 ……2
```

14は16進でEだから，
3624(10進)＝E28（16進）
というふうになります。
光：いちばんポピュラーな，人間向きのやり方ですね。
Yo：コンピュータ向きのやり方というのもあるの？
光：コンピュータ相手には掛け算がネックなんですよ。加算，減算とビットシフトで処理したほうがコンピュータには楽なんです。
純：どうすればいいんですか？
老：それぞれの桁の重みを引いていけばよいのじゃ。
Yo：1，2，4，8，16，32ってやつ？
老：そのとおりじゃ。それが何回引けたかを，上の桁から順番に格納していけばよい。16ビットの16進数を10進の文字列にするときは，最大の桁数が5桁（65535）じゃから，5桁分の重みのテーブルを10000，1000，100，10，1と作ってやればよいな。
M：納得できましたか？
純：できましたよ～んだ。
Yo：最初っから最後まで数字だったわね。頭が痛くなっちゃったわ。
光：乗除算もしっかり説明しようとすると結構大変ですね。
老：情報数学専攻してる数学オンチもいるがなぁ。みんなはここまでいわんでもわかるとは思うが。
で：ほっといてちょーだい。
M：残りの部分はどうするんですか？

純：メイン部分はまた来月払いということで……。

M：オヨヨ。今月の残りのサブルーチンの説明ぐらいはしてってくださいよ。

純：SEARCHSUBは演算子を判別させるための文字列判定ルーチン。そのほかのルーチンは見てのとおり，ワークからデータをPUSH, POPさせるものです。

Yo：あら？　そういえば善司君と（て）さんは？

M：さっきメアリーと帰っちゃいましたよ。

光：ぬわぁにーっ（がたたん）！

Yo：（ギロッ）そんなに心配？

光：えっ，いやーはっはっは，そんなことはぜーんぜん。じゃ，よーこさん，僕らも帰りましょうか。

Yo：私，お仕事残っているから。

光：そんなぁ〜。

老：さて，ワシらも帰るかのう。

純：へーい。さあさあ，光ちゃん帰りましょ。

M：ありがとうございました。来月こそツケは完済してくださいよ。

　カランコロ〜ン。

リスト1

```
0000                  1
0000                  2 ;DENTAKU in Z80' Bar  SUB
0000                  3 ;        1990.5.1 by Junji
0000                  4
0000                  5 #PRINT  EQU $1FF4
0000                  6 #PRINTS EQU $1FF1
0000                  7 #LTNL   EQU $1FEE
0000                  8 #MSG    EQU $1FE8
0000                  9 #MSX    EQU $1FE5
0000                 10 #MPRINT EQU $1FE2
0000                 11 #BELL   EQU $1FC4
0000                 12 #PRTHX  EQU $1FC1
0000                 13 #PRTHL  EQU $1FBE
0000                 14 #HLHEX  EQU $1FB2
0000                 15 #2HEX   EQU $1FB5
0000                 16 #FLGET  EQU $2021
0000                 17 #WIDCH  EQU $2030
0000                 18 #GETL   EQU $1FD3
0000                 19
9000                 20         ORG   $9000
9000                 21
9000 C3 55 92        22         JP    ENTRY
9003                 23
9003                 24 ;OUT ENZANSHI
9003                 25 OUTENZ
9003 4F              26         LD    C,A
9004 06 00           27         LD    B,00
9006 3E 02           28         LD    A,02
9008 CD 40 90        29         CALL  WPWRITE
900B C9              30         RET
900C                 31
900C                 32 ;PUSH DATA
900C                 33 PUSHD
900C ED 73 1A 90     34         LD    (PD+1),SP
9010 ED 7B 88 91     35         LD    SP,(CALADR)
9014 C5              36         PUSH  BC
9015 ED 73 88 91     37         LD    (CALADR),SP
9019 31 00 00        38 PD      LD    SP,0000
901C C9              39         RET
901D                 40
901D                 41 ;POP DATA
901D                 42 POPD
901D ED 73 2B 90     43         LD    (PD2+1),SP
9021 ED 7B 88 91     44         LD    SP,(CALADR)
9025 C1              45         POP   BC
9026 ED 73 88 91     46         LD    (CALADR),SP
902A 31 00 00        47 PD2     LD    SP,0000
902D C9              48         RET
902E                 49
902E                 50 ;ENZ PUSH
902E                 51 ENZPUSH
902E 2A 86 91        52         LD    HL,(ESPADR)
9031 23              53         INC   HL
9032 77              54         LD    (HL),A
9033 22 86 91        55         LD    (ESPADR),HL
9036 C9              56         RET
9037                 57
9037                 58 ENZPOP
9037 2A 86 91        59         LD    HL,(ESPADR)
903A 7E              60         LD    A,(HL)
903B 2B              61         DEC   HL
903C 22 86 91        62         LD    (ESPADR),HL
903F C9              63         RET
9040                 64
9040                 65 ;WORK PUSH
9040                 66 WPWRITE
9040 2A 84 91        67         LD    HL,(WPADR)
9043 77              68         LD    (HL),A
9044 23              69         INC   HL
9045 70              70         LD    (HL),B
9046 23              71         INC   HL
9047 71              72         LD    (HL),C
9048 23              73         INC   HL
9049 22 84 91        74         LD    (WPADR),HL
904C C9              75         RET
904D                 76
904D                 77 ;WORK POP
904D                 78 WPREAD
904D 2A 84 91        79         LD    HL,(WPADR)
9050 7E              80         LD    A,(HL)
9051 23              81         INC   HL
9052 46              82         LD    B,(HL)
9053 23              83         INC   HL
9054 4E              84         LD    C,(HL)
9055 23              85         INC   HL
9056 22 84 91        86         LD    (WPADR),HL
9059 C9              87         RET
905A                 88
905A                 89
905A                 90 ;文字列サーチ  DE=文字列ポインタ
905A                 91
905A                 92 SEARCHSUB
905A 06 00           93         LD    B,00        ;文字列番号
905C                 94 SEA4
905C ED 53 5E 91     95         LD    (INSW),DE
9060                 96 SEA3
9060 1A              97         LD    A,(DE)
9061 FE 20           98         CP    $20
9063 28 1E           99         JR    Z,SET?      ;文字列終了?
9065 FE 0D          100         CP    $0D
9067 28 1A          101         JR    Z,SET?
9069                102
9069 4F             103         LD    C,A
906A 7E             104         LD    A,(HL)
906B FE FF          105         CP    $FF
906D 28 1A          106         JR    Z,NOTERR
906F B7             107         OR    A
9070 C8             108         RET   Z           ;文字列番号セット
9071 B9             109         CP    C
9072 20 04          110         JR    NZ,SEA2
9074 23             111         INC   HL
9075 13             112         INC   DE
9076 18 E8          113         JR    SEA3
9078                114 SEA2
9078 CD 8B 90       115         CALL  SKIPSUB
907B                116 SEA5
907B 23             117         INC   HL
907C 04             118         INC   B
907D ED 5B 5E 91    119         LD    DE,(INSW)
9081 18 D9          120         JR    SEA4
9083                121 SET?
9083 7E             122         LD    A,(HL)      ;END CODE CHECK
9084 B7             123         OR    A
9085 20 F1          124         JR    NZ,SEA2
9087 B7             125         OR    A
9088 C9             126         RET
9089                127 NOTERR
9089 37             128         SCF
908A C9             129         RET
908B                130
908B                131 ;文字列スキップ
908B                132
908B                133 SKIPSUB
908B 7E             134         LD    A,(HL)
908C B7             135         OR    A
908D C8             136         RET   Z
908E 23             137         INC   HL
908F 18 FA          138         JR    SKIPSUB
9091                139
9091                140 ;加算  HL=HL+DE
9091                141 PLUS
9091 19             142         ADD   HL,DE
9092 C9             143         RET
9093                144
9093                145 ;減算  HL=HL-DE
9093                146 MINUS
9093 B7             147         OR    A
9094 ED 52          148         SBC   HL,DE
9096 C9             149         RET
9097                150
9097                151 ;乗算    HL=HL*DE
9097                152 MULTI16
9097 4B             153         LD    C,E
9098 42             154         LD    B,D
9099 11 00 00       155         LD    DE,0000
909C 3E 10          156         LD    A,16
909E                157 MUL3
909E CB 38          158         SRL   B
90A0 CB 19          159         RR    C
90A2 30 03          160         JR    NC,MUL2
90A4 EB             161         EX    DE,HL
90A5 19             162         ADD   HL,DE
90A6 EB             163         EX    DE,HL
90A7                164 MUL2
90A7 29             165         ADD   HL,HL
90A8 3D             166         DEC   A
90A9 20 F3          167         JR    NZ,MUL3
90AB EB             168         EX    DE,HL
90AC C9             169         RET
90AD                170
90AD                171 ;除算      HL/DE,HL=ANS,DE=AMARI
90AD                172 DIV16
90AD C5             173         PUSH  BC
90AE 3E 10          174         LD    A,16
```

```
90B0 4B           175             LD      C,E
90B1 42           176             LD      B,D
90B2 EB           177             EX      DE,HL
90B3 21 00 00     178             LD      HL,0000
90B6              179 DIV162
90B6 29           180             ADD     HL,HL
90B7 EB           181             EX      DE,HL
90B8 29           182             ADD     HL,HL
90B9 EB           183             EX      DE,HL
90BA 30 01        184             JR      NC,DIV163
90BC 23           185             INC     HL
90BD              186 DIV163
90BD B7           187             OR      A
90BE ED 42        188             SBC     HL,BC
90C0 30 03        189             JR      NC,DIV164
90C2 09           190             ADD     HL,BC
90C3 18 01        191             JR      DIV165
90C5              192 DIV164
90C5 13           193             INC     DE
90C6              194 DIV165
90C6 3D           195             DEC     A
90C7 20 ED        196             JR      NZ,DIV162
90C9 EB           197             EX      DE,HL
90CA C1           198             POP     BC
90CB C9           199             RET
90CC              200
90CC              201 MOD16
90CC CD AD 90     202             CALL    DIV16
90CF EB           203             EX      DE,HL
90D0 C9           204             RET
90D1              205
90D1              206 ;10進文字列を数値に変換
90D1              207
90D1              208 NUM10
90D1 ED 53 60 91  209             LD      (NUMWORK),DE
90D5 21 00 00     210             LD      HL,0000
90D8              211 NUM2
90D8 29           212             ADD     HL,HL
90D9 4D           213             LD      C,L
90DA 44           214             LD      B,H
90DB 29           215             ADD     HL,HL
90DC 29           216             ADD     HL,HL
90DD 09           217             ADD     HL,BC
90DE              218
90DE 1A           219             LD      A,(DE)
90DF CD 02 91     220             CALL    CHECK
90E2 38 0F        221             JR      C,NUMCHK
90E4 13           222             INC     DE
90E5 D6 30        223             SUB     "0"
90E7 4F           224             LD      C,A
90E8 06 00        225             LD      B,00
90EA 09           226             ADD     HL,BC
90EB 1A           227             LD      A,(DE)
90EC CD 02 91     228             CALL    CHECK
90EF 30 E7        229             JR      NC,NUM2
90F1 B7           230             OR      A
90F2 C9           231             RET
90F3              232 NUMCHK
90F3 E5           233             PUSH    HL
90F4 2A 60 91     234             LD      HL,(NUMWORK)
90F7 B7           235             OR      A
90F8 ED 52        236             SBC     HL,DE
90FA E1           237             POP     HL
90FB CA 00 91     238             JP      Z,NUMERR
90FE B7           239             OR      A
90FF C9           240             RET
9100              241 NUMERR
9100 37           242             SCF
9101 C9           243             RET
9102              244
9102              245 CHECK
9102 FE 3A        246             CP      "9"+1
9104 38 02        247             JR      C,CHECK2
9106              248 CHKEND
9106 37           249             SCF
9107 C9           250             RET
9108              251 CHECK2
9108 FE 30        252             CP      "0"
910A C9           253             RET
910B              254
910B              255 ;数値を10進文字列に変換
910B              256
910B              257 STRING16
910B CD 22 91     258             CALL    STRSUB
910E 11 58 91     259             LD      DE,STRD16
9111 06 04        260             LD      B,04
9113              261 STR163
9113 1A           262             LD      A,(DE)
9114 FE 30        263             CP      "0"
9116 20 06        264             JR      NZ,STR162
9118 CD F1 1F     265             CALL    #PRINTS
911B 13           266             INC     DE
911C 10 F5        267             DJNZ    STR163
911E              268 STR162
911E CD E5 1F     269             CALL    #MSX
9121 C9           270             RET
9122              271
9122              272 STRSUB
9122 D9           273             EXX
9123 21 58 91     274             LD      HL,STRD16
9126 D9           275             EXX
9127 11 4E 91     276             LD      DE,DECTBL
912A 3E 05        277             LD      A,05
912C              278 STRLOOP
912C EB           279             EX      DE,HL
912D 4E           280             LD      C,(HL)
912E 23           281             INC     HL
912F 46           282             LD      B,(HL)
9130 23           283             INC     HL
9131 EB           284             EX      DE,HL
9132 08           285             EX      AF,AF'
9133 97           286             SUB     A
9134 08           287             EX      AF,AF'
9135              288 SLOOP2
9135 B7           289             OR      A
9136 ED 42        290             SBC     HL,BC
9138 38 05        291             JR      C,STRSUB2
913A 08           292             EX      AF,AF'
913B 3C           293             INC     A
913C 08           294             EX      AF,AF'
913D 18 F6        295             JR      SLOOP2
913F              296 STRSUB2
913F 09           297             ADD     HL,BC
9140 08           298             EX      AF,AF'
9141 C6 30        299             ADD     A,"0"
9143 D9           300             EXX
9144 77           301             LD      (HL),A
9145 23           302             INC     HL
9146 D9           303             EXX
9147 08           304             EX      AF,AF'
9148 3D           305             DEC     A
9149 20 E1        306             JR      NZ,STRLOOP
914B 97           307             SUB     A
914C 77           308             LD      (HL),A      ;END CODE
914D C9           309             RET
914E              310
914E 10 27 E8 03  311 DECTBL  DW      10000,1000,100,10,1
9152 64 00 0A 00
9156 01 00
9158 00 00 00 00  312 STRD16  DS      05
915C 00
915D 00           313         DB      00
915E 00 00        314 INSW    DW      0000
9160 00 00        315 NUMWORK DW      0000
9162              316
9162              317 CALTBL
9162 2A 00        318         DB      "*",00
9164 2F 00        319         DB      "/",00
9166 44 00        320         DB      "MOD",00
9168 2B 00        321         DB      "+",00
916A 2D 00        322         DB      "-",00
916C 28 00        323         DB      "(",00
916E FF           324         DB      $FF
916F              325
916F              326 JUMPTBL
916F 97 90        327         DW      MULTI16
9171 AD 90        328         DW      DIV16
9173 CC 90        329         DW      MOD16
9175 91 90        330         DW      PLUS
9177 93 90        331         DW      MINUS
9179              332
9179 00 00 00 00  333 MEMDAT  DS      10
917D 00 00 00 00
9181 00 00
9183 00           334 MEMNO.  DB      00
9184 04 92        335 WPADR   DW      WPSP
9186 40 92        336 ESPADR  DW      ENZSP
9188 04 92        337 CALADR  DW      CALSP
918A 00 00        338 ANSWER  DW      0000
918C              339
918C 00 00 00 00  340 LIGET   DS      80
9190 00 00 00 00
9194 00 00 00 00
9198 00 00 00 00
919C 00 00 00 00
91A0 00 00 00 00
91A4 00 00 00 00
91A8 00 00 00 00
91AC 00 00 00 00
91B0 00 00 00 00
91B4 00 00 00 00
91B8 00 00 00 00
91BC 00 00 00 00
91C0 00 00 00 00
91C4 00 00 00 00
91C8 00 00 00 00
91CC 00 00 00 00
91D0 00 00 00 00
91D4 00 00 00 00
91D8 00 00 00 00
91DC 00 00 00 00  341         DS      40
91E0 00 00 00 00
91E4 00 00 00 00
91E8 00 00 00 00
91EC 00 00 00 00
91F0 00 00 00 00
91F4 00 00 00 00
91F8 00 00 00 00
91FC 00 00 00 00
9200 00 00 00 00
9204              342 CALSP
9204 00 00 00 00  343 WPSP    DS      3*20
9208 00 00 00 00
920C 00 00 00 00
9210 00 00 00 00
9214 00 00 00 00
9218 00 00 00 00
921C 00 00 00 00
9220 00 00 00 00
9224 00 00 00 00
9228 00 00 00 00
922C 00 00 00 00
9230 00 00 00 00
9234 00 00 00 00
9238 00 00 00 00
923C 00 00 00 00
9240 07           344 ENZSP   DB      $07
9241 00 00 00 00  345         DS      20
9245 00 00 00 00
9249 00 00 00 00
924D 00 00 00 00
9251 00 00 00 00
9255              346
9255              347 SUBEND
9255              348
9255              349
9255              350
9255              351
9255              352
```

X68000用　©日本テレネット

夢幻戦士ヴァリスIIより

SACRED SACRIFICE

Watanabe Kazuhiko
渡辺 一彦

X1/turbo用

トッカータとフーガ二短調BWV565

Hanai Akitada
花井 章能

毎日雨模様が続いていますが，皆さんいかがお過ごしでしょうか。さて，今月は予定を変更して，X68000用には夢幻戦士ヴァリスIIを，X1用にはトッカータとフーガをお送りします。しとしとと降るやさしい雨を眺めながら，たまにはじっくりとクラシックに浸ってみるのもいいもんですよ。

そろそろ巷では1学期期末試験などという，“恐怖のセレモニー”が待ち受けている頃でしょう。普段は“あの頃に戻りたい”などと言っている人も，この時期だけには戻りたくはありませんよね。ノートを借りまくっている人もいることでしょう。皆さん，いかがお過ごしでしょうか？　私はコピー機を買いました。それから，先月号で予告していたT-SQUAREの“OMENS OF LOVE”は，ページの都合で来月回しとなってしまいました。申し訳ありません。自分の作品かな？　って期待していた人，本当にゴメンなさい。来月号まで待っていてくださいね。

優子はいけにえ？

夢幻戦士ヴァリスIIより，オープニングテーマ「SACRED SACRIFICE」をお届けしましょう。直訳すると「聖なる生贄」と読めそうです。そうか，優子ちゃんは生贄だったのか……とひとりで納得していた私です。ちなみにX68000用のヴァリスIIではミュージックモードがあり，私服姿の優子ちゃんが見られます。ミュージックモードへの入り方は，オープニングデモの途中でF5キーを押してみてください。原曲と聴き比べるとわかりますが，渡辺君の作品はかなりスマートに仕上がっています。効果音ビシバシ，ヴォリューム目一杯で怒鳴っているような（失礼）あのオープニングテーマが，かなり洗練されています。騒ぐところは騒いで，抑えるところは抑えるという，メリハリが効いているのが素晴らしいでしょう。初投稿なのにこのレベルとは「なかなかやるわい」級といったところですね。ところで，原曲からしてヘビメタぽかったのに，作者の趣味に走ってしまって完全にそっちの世界に踏み外してるような気もしますが，大歓迎です。別にメタルじゃなくたっていいんです，完全に自分の世界に逃避してしまったようなアレンジバージョンも歓迎するということで，そちらのスジの方もお待ち申し上げております。

失われた惑星

X1用には「トッカータとフーガ二短調BWV565」をお届けしましょう。この曲はクラッシックのなかでも，超ポピュラーな部類に入ると思えるくらい有名ですので，タイトルは知らない人でも，一度は聴いたことがあると思います。そういえば私の頼りない記憶では，幻魔大戦のサントラにも「失われた惑星」というタイトルで収録されてたと思います。こちらのほうは，フルコーラスではなかったように記憶していますが……。この方面に詳しい人は，教えてくださいね。

さて，作品の完成度なのですが，うーむ。全体的にはとってもよくできています。入力して損したと思うことはまずないでしょう。エコーの残し具合や音の響き具合，間合いなどは苦心のあとがうかがえます。作者の花井君はよほど誰かの（演奏した）曲を聴きこんだのでしょう。かなり統一されたイメージで演奏されます。そういった面ですばらしかったからこそ，このページに載っていると思ってください。

夢幻戦士ヴァリスII

ところで，問題点としては，部分的にプチノイズが残っていたりして，せっかくの曲の完成度を曇らせているところなのです。それ以外の出来のことを考えると，ちょっともったいないかな，って気がします。しょうがないと言ってしまえばそれまでなのですが，やはりどうにかしたいというのが人心というものです。これは，音色の変更で軽減する可能性がありますので，各自研究してみてください。

LIVE inではこのページに対する希望やリクエスト，質問などもお待ちしております。アンケート葉書や，投稿作品に同封していただく手紙などに書いてくれたらうれしいなっと。　(S.K.)

リスト1　夢幻戦士ヴァリスII

```
 10 /*
 20 /*          T h e   F a n t a s m   S o l d e r
 30 /*                   夢幻戦士ヴァリス ][
 40 /*
 50 /*
 60 /*               O P E N I N G   T h e m e
 70 /*           S A C R E D   S A C R I F I C E
 80 /*
 90 /*      For OPMA 89/11/11 Sat.   I p p i k o . W
100 /* CDの ｱﾚﾝｼﾞﾊﾞｰｼﾞｮﾝをもとに，かってにじぶんなりのｱﾚﾝｼﾞをしちゃたのよよーんんんんん
110 m_init()
120 dim char v(4,10) : str m(99)[255],d(99)[99],sd1[10],sd2[10
],sd3[10],bd[10],ht[10],mt[10],lt[10]
130   /*
140   for i=1 to 8
150       m_alloc(i,5000)
160           m_assign(i,i)
170       next  : m_alloc(9,32)
180 /* Drums Set
190 sd1="y2,14
200 sd2="y2,17
210 sd3="y2,21
```

```
220  bd="y2,23
230 /*
240      ht="y2,24
250      mt="y2,25
260      lt="y2,26
270 /* Elec. Tom
280  d(0)=" {b&b-&a&a-&g&g-&f&e}8 "
290  d(1)=" {b&b-&a&a-}16
300  d(2)=" {b&b-}32
310 /*
320 d(10)=" {g&g-&f&e&e-&d&d-&c}8
330 d(11)=" {g&g-&f&e}16
340 d(12)=" {g&g-}32
350 /*
360 d(20)=" {e&e-&d&d-&c&>b&b-&a}8
370 d(21)=" {e&e-&d&d-}16
380 d(22)=" {e&e-}32
390 /*
400 d(30)=" {c&>b&b-&a&a-&g&g-&f}8
410 d(31)=" {c&>b&b-&a}16
420 d(32)=" {c&>b}32
430 /* MML Data
440 v={
450 /*      A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
460          56, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
470 /*       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
480          17, 16,  0,  3,  3, 26,  0,  3,  3,  0,  0,
490          20,  0,  0,  3,  0, 27,  0,  1,  0,  0,  0,
500          28,  0,  0,  3,  0, 28,  0,  1,  0,  0,  0,
510          28,  0,  0,  8,  0,  0,  0,  1,  4,  0,  0}
520   m_vset(70,v)              :/* Elec. Guiter
530 /*
540 v={
550 /*      A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
560          57, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
570 /*       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
580          31, 11,  0,  2,  2, 36,  2,  9,  0,  0,  0,
590          31,  5,  0,  2,  1, 20,  1,  0,  0,  0,  0,
600          31, 10,  5,  5,  1, 34,  1,  0,  0,  0,  0,
610          31,  5,  2,  8,  1,  4,  1,  0,  0,  0,  0}
620   m_vset(71,v)              :/* Bass
630 /*
640 v={
650 /*      A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
660          59, 15,  2,  1,  0,  0,  0,  0,  0,  3,  0,
670 /*       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
680          20,  0,  0,  2,  0, 15,  0,  7,  4,  0,  0,
690          31, 14,  0,  2,  2, 25,  0,  3,  0,  0,  0,
700          31, 12,  0,  2,  2, 20,  0,  4,  2,  0,  0,
710          31,  0,  0,  6,  0,  0,  0,  5,  7,  0,  0}
720   m_vset(72,v)              :/* Soft Dist. Guiter
730 /*
740 v={
750 /*      A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
760          62, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
770 /*       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
780          31,  5,  5,  5,  2,  7,  0,  0,  1,  0,  0,
790          31, 16, 11,  8, 13,  0,  0,  0,  0,  0,  0,
800          31, 17, 10,  8, 12,  0,  0,  0,  0,  0,  0,
810          31, 12, 10,  8, 12,  0,  0,  0,  0,  0,  0}
820   m_vset(73,v)              :/* Tom Tom
830 /*
840 v={
850 /*      A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
860          60, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0,
870 /*       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
880          31, 10,  8, 15, 15,  0,  0, 12,  3,  1,  0,
890          31, 17, 10, 15, 15,  0,  0,  8,  3,  0,  0,
900          31, 10,  8, 15, 15,  0,  0, 12,  7,  1,  0,
910          31, 17, 10, 15, 15,  0,  0,  8,  3,  0,  0}
920   m_vset(68,v)                :/* Hi-Hat close
930 /*
940 v={
950 /*      A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
960          56, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0,
970 /*       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
980          26,  0,  8,  5,  3, 22,  0,  8,  4,  3,  0,
990          31, 11,  8,  7,  0, 30,  0,  8,  0,  2,  0,
1000         31, 11,  8,  7,  0, 25,  0,  8,  0,  1,  0,
1010         31, 17, 10,  7,  3,  0,  0,  4,  0,  0,  0}
1020  m_vset(82,v)                :/* Hi - Hat Open
1030 /*
1040 v={
1050 /*     A/F  OM  WF  SY  SP PMD AMD PMS AMS PAN nothing
1060         59, 15,  2,  0,200, 60,  0,  3,  0,  3,  0,
1070 /*      AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
1080         31,  4,  2,  2, 15, 20,  0,  1,  4,  1,  0,
1090         31,  2,  0,  1, 15, 28,  0,  4,  0,  2,  0,
1100         31,  8,  0,  4, 15, 35,  0, 11,  0,  2,  0,
1110         31, 12,  4,  6, 15,  0,  0,  1,  3,  0,  0}
1120  m_vset(83,v)                :/* Ride Cymb
1130 /*
1140 m_tempo(132) :/* Track Data
1150 /* track 1 (Main)
1160 m(0)=" r8 [do] y48,30 @70 p3 o4 v14
1170 m(1)=" 16 |: e-de-fe-d :| d8r8d8c8&c1&c2
1180 m(2)=" {g&g#&a&a#&b&<c&c#&d&e-&}16&e-4..&e-1 |: {e-&e&f&f#
&g&}16&g4&g16 |1 d1&d2&d8  :| |2 c1&c8 {cde-fe-d}1
1190 m(3)=" c2..{>b<c>}16b16& b1
1200 m(4)=" c1 d1
1210 m(5)=" v15 |: p3 o4 g1 g1 f1 f1 e-1 e-1 d1 p1 g16g16r4g16g
16r4r4 :|
1220 m(99)="[loop]
1230 m_trk(1,m(0))
1240 m_trk(1,m(1))
1250 m_trk(1,m(1))
1260 m_trk(1,m(1))
1270 m_trk(1,m(1))
1280 m_trk(1,"o3"+m(2))
1290 m_trk(1,m(3))
1300 m_trk(1,m(2))
1310 m_trk(1,m(4))
1320 m_trk(1,m(5))
1330 m_trk(1,m(99))
1340 /* track 2 (Main ﾊﾓﾘ)
1350 m(0)=" y48,10 r8 [do] @70 p3 o4 v12 r8
1360 m(4)=" c1 d2.. v12
1370 m(5)=" v15 |: p3 o4 g1 p1 r4e-2. p3 f1 p1 r4d2. p3 e-1 p1
r4c2. d1 < p2 g16g16r4g16g16r4r4 > :|
1380 m_trk(2,m(0))
1390 m_trk(2,m(1))
1400 m_trk(2,m(1))
1410 m_trk(2,m(1))
1420 m_trk(2,m(1))
1430 m_trk(2,"o3"+m(2))
1440 m_trk(2,m(3))
1450 m_trk(2,m(2))
1460 m_trk(2,m(4))
1470 m_trk(2,m(5))
1480 m_trk(2,m(99))
1490 /* track 2 (Sub)
1500 m(0)=" y50,30 r8 [do] @70 p1 o4 v13
1510 m(1)=" 16 |: c>b-<c cc>b-< :| >b-8r8b-8a-8&a-2&a-1 <
1520 m(4)=" c1 d1
1530 m(5)=" v15 |: p2 o4 r1 r2c2 > r1 r2b-2 r1 r2a-2 r1 p2 < g1
6g16r4g16g16r4r4  :|
1540 m_trk(3,m(0))
1550 m_trk(3,m(1))
1560 m_trk(3,m(1))
1570 m_trk(3,m(1))
1580 m_trk(3,m(1))
1590 m_trk(3,"o3 v12 "+m(2))
1600 m_trk(3,m(3))
1610 m_trk(3,m(2))
1620 m_trk(3,m(4))
1630 m_trk(3,m(5))
1640 m_trk(3,m(99))
1650 /* track 2 (Sub ﾊﾓﾘ)
1660 m(0)=" y51,10 r8 [do] @70 p2 o4 v11 r8
1670 m(4)=" c1 d2.. v12
1680 m(5)=" v15 |: p3 o3 r1 r2.a-4 r1 r2.g4 r1 r2.f4 r1 p1 < d1
6d16r4d16d16r4r4> :|
1690 m_trk(4,m(0))
1700 m_trk(4,m(1))
1710 m_trk(4,m(1))
1720 m_trk(4,m(1))
1730 m_trk(4,m(1))
1740 m_trk(4,"o3 v12"+m(2))
1750 m_trk(4,m(3))
1760 m_trk(4,m(2))
1770 m_trk(4,m(4))
1780 m_trk(4,m(5))
1790 m_trk(4,m(99))
1800 /* track 2 (bass)
1810 m(0)=" @71 l8 v13 o2 y3,3 y2,23g16y2,23g16 [do] o3
1820 m(1)=" c2..> g16g16 <c1 > a-1& a-1 < |:4 cc16c16c16c16c8 :
| > |: a-a-16a-16a-16a-16a- :| g16<e->a-16<d>a16<d->b-16<c> {gab
-g}4 < |: |:4 cc16c16c16c16c :| > |:4 a-a-16a-16a-16a-16a- :| <
|:4 cc16c16c16c16c :| > |:4 a-a-16a-16a-16a-16a- :|
1830 m(2)=" < |:4 c{cccc}4c :| > |:4 b-{b-b-b-b-}4b- :| |:4 a-{
a-a-a-a-}4a- :| |: g{gggg}4g :| g{ggag}4g g{gggg}4g <
1840 m(3)=" |:4 c{cccc}4c :| > |:4 b-{b-b-b-b-}4b- :| |:4 a-{a-
a-a-a-}4a- :| |:4 g{gggg}4g :|
1850 m(4)=" |: a-a-a-a-a-a-a-a- a-a-a-a-a-a-a-a- gggggggg ggggg
ggg a-a-a-a-a-a-a-a- a-a-a-a-a-a-a-a- gggggggg g16g16r8 <g8>g16g
16 r8<g8> {gab-g}4 :|
1860 m_trk(5,m(0))
1870 m_trk(5,m(1))
1880 m_trk(5,m(2))
1890 m_trk(5,m(3))
1900 m_trk(5,m(4))
1910 m_trk(5,m(99))
1920 /* track 6 (Distotion Guitar)
1930 m(0)=" y53,60 @72 o2 v12 q7 g16g16 [do] o3 l8
1940 /*m(2)=" < c1&c1 > b-1&b-1 a-1&a-1 g1&g1
1950 m(2)=" < |:4 c{cccc}4c :| > |:4 b-{b-b-b-b-}4b- :| |:4 a-{
a-a-a-a-}4a- :| |:4 g{gggg}4g :|
1960 m_trk(6,m(0))
1970 m_trk(6,m(1))
1980 m_trk(6,m(2))
1990 m_trk(6,m(2))
2000 m_trk(6,m(4))
2010 m_trk(6,m(99))
2020 /* track 7 (Distotion Guitar ﾊﾓﾘ)
2030 m(0)=" y54,00 @72 o1 v12 q7 g16g16 [do] o2 l8
2040 m_trk(7,m(0))
2050 m_trk(7,m(1))
2060 m_trk(7,m(2))
2070 m_trk(7,m(2))
2080 m_trk(7,m(4))
2090 m_trk(7,m(99))
2100 /* Track 8 (Drums)
2110 m(0)=" r8 [do] y3,3 @v127 l8 o4
2120 m(1)=" y2,5r2..|:"+bd+"r16:| |: y3,3"+ht+"r48"+ht+"r12..y3
,1"+mt+"r48"+mt+"r12..y3,2"+lt+"r48"+lt+"r12..:| y3,3 |:12"+bd+"
r :| r4 y3,1 |:4 y2,62 r32 :| y3,2 |:4 y2,64 r32 :|
2130 m(2)=" y3,3 @83 p1"+bd+"b"+sd3+"b16&"+bd+"b16 b16&"+bd+"b1
6"+sd3+"b :|
2140 m(3)=bd+"r16"+"y2,24r64y2,24r16..y3,3"+bd+"r16"+"y3,1y2,25
r64y2,25r16..y3,3  "+bd+"r16 y3,2y2,26r64y2,26r16..y3,3"+bd+"r16
```


▶部屋のレイアウトを変えた。Oh！X とアシベも買った。人，これを逃避行動という。中間考査まであと３日。こんなときに付録付きで出すなんざ小粋なことをしてくれるぜ！

佐藤　雄二（17）新潟県

```
y2,27r64y2,27r16.. |:4"+sd1+"r32 :| |:"+sd1+"r16:|
2150 m(4)=bd+"b"+sd3+"b16&"+bd+"b16 b16& |:"+sd3+"b16& :|"+sd3+
"b16
2160 m(5)=" |:4 "+sd3+"b16&"+sd3+"b16 :|
2170 m(6)=" l16 @68 p2 o4 cccc @68 p2 cc @83 p2 f8 |:"+bd+"@68
p2 cc @83 p2 f8 :|
2180 m(12)="l16 @68 p2 o4 cccc |:3"+bd+"@68 p2 cc @83 p2 f8 :|
2190 m(7)=" @68 p2 o4 cccc  |:2 "+bd+"@68 p2 cc @83 p2 f8 :| @6
8 p2 |:"+sd3+"c :| @83 p2 "+sd3+"f& "+sd3+"f
2200 m(8)=" @68 p2 o4 cccc "+bd+"@68 p2 cc @83 p2 f8  @68 p2 |:
"+sd3+"c :| @83 p2 y3,1"+ht+"f&"+ht+"f @68 p2 y3,2 |:"+mt+"c :|
@83 p2 y3,3 "+lt+"f&"+lt+"f
2210 m(9)="@68p2y2,5cccc"+sd1+"cc@83p2f8@68p2"+bd+"ccy3,1"+ht+"
c64&"+ht+"c32.y3,2"+lt+"cy3,3"+sd1+"cc@83p2f8|:@68p2"+bd+"cccc"+
sd1+"cc@83p2f8@68p2"+bd+"ccy3,1"+ht+"c64&"+ht+"c32.y3,2"+lt+"cy3
,3"+sd1+"cc@83p2f8:|
2220 m(10)="@68 p2"+bd+"cccc"+sd1+"cc@83p2f8@68p2"+bd+"cc"+mt+"
c64&"+mt+"c32.c|:"+sd1+"c:|@83p2"+sd1+"f&"+sd1+"f|:2@68p2"+bd+"c
ccc"+sd1+"cc@83p2f8@68p2"+bd+"ccy3,1"+ht+"c64&"+ht+"c32.y3,2"+lt
+"cy3,3"+sd1+"cc@83p2f8:|
2230 m(11)=bd+"@68 p2 cccc "+sd1+"cc @83 p2 f8 @68 p2"+bd+"cc |
: y2,63 c32 :| |:3 y2,63 c :| @83 p2 |:y2,63 f& :|  |:"+bd+"r16
:| r8 y2,5 r8 |:"+bd+"r16 :| r8 y2,5 r8
2240 m(13)="y3,3 y2,62 r y2,62 r y3,1 y2,63 r y3,2 y2,64 r y3,3
2250 m(14)=bd+"|:"+sd1+"r32 :|"+sd1+"r16"+sd1+"r8
2260 m_trk(8,m(0))
2270 m_trk(8,m(1))
2280 m_trk(8,"|:6"+m(2))
2290 m_trk(8,m(3))
2300 m_trk(8,"|:7"+m(2))
2310 m_trk(8,m(4))
2320 m_trk(8,"|:7"+m(2))
2330 m_trk(8,m(5))
2340 m_trk(8,"y2,44"+m(6))
2350 m_trk(8,bd+m(12))
2360 m_trk(8,"y2,43"+m(6))
2370 m_trk(8,bd+m(12))
2380 m_trk(8,"y2,41"+m(6))
2390 m_trk(8,bd+m(12))
2400 m_trk(8,"y2,41"+m(6))
2410 m_trk(8,bd+m(7))
2420 m_trk(8,"y2,44"+m(6))
2430 m_trk(8,bd+m(12))
2440 m_trk(8,"y2,43"+m(6))
2450 m_trk(8,bd+m(12))
2460 m_trk(8,"y2,41"+m(6))
2470 m_trk(8,bd+m(12))
2480 m_trk(8,"y2,41"+m(6))
2490 m_trk(8,bd+m(8))
2500 m_trk(8,m(9))
2510 m_trk(8,m(10))
2520 m_trk(8,m(11)+m(13))
2530 m_trk(8,m(9))
2540 m_trk(8,m(10))
2550 m_trk(8,m(11)+m(14))
2560 m_trk(8,m(99))
2570 for i=1 to 8
2580     m_assign(i,i)
2590         next
2600 print "Push Space bar to start!! " : while inkey$<>" "
2610     endwhile
2620         m_play()
2630     end
```

リスト2 トッカータとフーガ1

```
10 REM ***************************************
20 REM *                                     *
30 REM *   TOCCATA and FUGUE in D MINOR      *
40 REM *                                     *
50 REM *                     by J.S.BACH     *
60 REM *                                     *
70 REM ***************************************
80 TEMPO 0
90 DEFSTR A-N:DEFINT O-Z
100 DIM A(20),B(20),C(20),D(20),E(20),F(20),G(20),H(20)
110 MEM$(&HB190,36)=HEXCHR$("FD000204010028060404lF1212121482828
2000202020008080800000000000080000000")
120 MEM$(&HB1B4,36)=HEXCHR$("FD000204010028060404lF1212121482828
2000101010005050500000000000080000000")
130 GOTO 200
140 '
150 LABEL "!"
160 A(R)=A:B(R)=B:C(R)=C:D(R)=D:E(R)=E:F(R)=F:G(R)=G:H(R)=H
170 R=R+1
180 RETURN
190 '
200 R=0
210 I="L16A32.G32.A2.R4.GFEDC+8.RD4.R1"
220 A="T120V11I1Q8P1K05 O6"+I
230 B="V11I1Q8P2K08 O6R64"+I
240 C="V04I2Q8P3K10 O6R16"+I
250 D="V03I2Q8P3K15 O6R08"+I
260 E="V11I1Q8P1K05 O5"+I
270 F="V11I1Q8P2K08 O5R64"+I
280 G="V04I2Q8P3K10 O5R16"+I
290 H="V03I2Q8P3K15 O5R08"+I
300 "!"
310 A="<A32.G32.A2.R4E8.F8.C+8.D4.R1<"
320 B=A:C=A:D=A:E=A:F=A:G=A:H=A
330 "!"
340 A=I+"R8":E=A
350 B=I+"R@21":F=B
360 C=I+"R16":G=C
370 D=I:H=D
380 "!"
390 A="T110V14I2P3K05 O2D1&D1&D1&D1&D1&D4R1"
400 B="V11I1P3K5 O4R1C+1&C+1&C+4 V12D1&D1R1"
410 C="V09I1P2K8 O4R1R32C+2.B-@90 V11P3K5>E1 V12D1&D1R1"
420 D="V11I1P3K5 O4R1R4E1&E1 V7I2P3K8R32D1&D1R@186"
430 E="V09I1P2K8 O4R1R4R32E@138 V11P3K5>C+1&C+4 <V12A1&A1R1"
440 F="V11I1P3K5 O4R1R2G1&G2. V12G4.E4.F+1&F+4R1"
450 G="V09I1P2K8 O4R1R2R32G2>C+4E1 V7I2P3D1&D1R@186"
460 H="V11I1P3K5 O4R1R2.B-1&B-2 V7I2K8R32A4.E4.F+1&F+4R@186"
470 "!"
480 I=">L16C+.DEC+DEC+DEC+D.EFGEFGEFGEF.GAB-GAB-GAB-GA4R1"
490 A="T145V09I1P1K05 O4"+I+I
500 B="V09I1P2K08 O4R64"+I+I
510 C="V04I2P3K10 O4R16"+I+I
520 D="V03I2P3K15 O4R08"+I+I
530 E="V09I1P1K05 O3"+I+I
540 F="V09I1P2K08 O3R64"+I+I
550 G="V04I2P3K10 O3R16"+I+I
560 H="V03I2P3K15 O3R08"+I+I
570 "!"
580 I="A.GB-EGB-EFADFADEGCEGCDF<B->DF<B->CE<A>CE<AB->D<GB->D<GA>
C<FA>C<FGB-EGB-EFADFADEGC+.E.G.C+8R8"
590 A="T150"+I:B=I:C=I:D=I:E=I:F=I:G=I:H=I
600 "!"
610 A="T135V14I2P3K5 O2D1&D1&D1&D2R2"
620 B="V11I2P3K5 O4R@189C+1&C+1&C+2R2"
630 C="V09I2P2K8 O4R@186C+2.B-@90 V11P3K5>E1&E4R2"
640 D="V11I2P3K5 O4R1R8E1&E4>G1R2"
650 E="V09I2P2K8 O4R1R4R32E@138 V11P3K5>C+1&C+2R2"
660 F="V11I2P3K5 O4R1R@93G1&G1R2"
670 G="V09I2P2K8 O4R1R@90G2>C+4E4G4B-1&B-@42"
680 H="V11I2P3K5 O4R1R2R8B-1>P1B-1&B-4"
690 "!"
700 I="L16AGFEDC+<B>C+8<A8>C+8EG.L32.FGFGFGF4R8"
710 A="V11I1P1K05 O5"+I+"E8."
720 B="V11I1P2K08 O5R64"+I+"E@33"
730 C="V04I2P3K10 O5R16"+I+"E8"
740 D="V03I2P3K15 O5R08"+I+"E16"
750 E="R1R@156":F=E:G=E:H=E
760 "!"
770 A="V12I1P1K5 O5F1&F1&F2&F32"
780 B="V12I2P2K8 O5R32F1&F1&F2"
790 C="V12I2P3K5 O5D1&D1&D2&D32"
800 D="V12I1P1K5 O4A1&A1&A2&A32"
810 E="V12I2P2K8 O4R32A1&A1&A2"
820 F="V12I2P3K5 O4F1&F1&F2&F32"
830 G="V12I2P3K5 O4D1&D1&D2&D32"
840 H="V14I2P3K5 O2D1&D1&D2&D32"
850 "!"
860 I="R1R2O5L8AQ3>DEFDEFGEFGAFGAB-GAFGEFDEC+D<AB-GAFGEFDGEFDEC+
D<AB-GAFGEFDGEFDEC+Q8D4R@60"
870 A="T130V10I1P1K05"+I
880 B="V10I1P2K08R64"+I
890 C="V04I2P3K10R16"+I
900 D="V03I2P3K15R08"+I
910 I="R1R2Q4O5R8.L8"+STRING$(56,"A")+"R2"
920 E="V08I1P1K05"+I
930 F="V08I1P2K08R64"+I
940 G="V04I2P3K10R16"+I
950 H="V03I2P3K15R08"+I
960 "!"
970 I="L16Q8DFB-FCEAE<B->DGD<A>C+EAQ7L8DB-<A>A<B->G"
980 A="T125O5"+I+"A4"+I+"Q8 A2."
990 B="O5"+I+"@27 V8I2P3A@45Q8 V10I1P2"+I+"@27Q8 V8I2P3A@141"
1000 C="O5"+I+"16 V10I1P1K5<E@60>Q8 V4I2P3K10"+I+"16Q8 V10I1P1K5
<E2R4>"
1010 D="O5"+I+"32 V8I2P3K8<E@66>Q8 V3K15"+I+"32Q8 V8K8<E@90R4>"
1020 E="V10R1Q7O5L8RFRERDC+4R1RFRERDQ8C+2R4"
1030 F="V10R1Q7O5L8RFRERD@27 V8I2P3C+@45 V10I1P2R1RFRERD@27Q8 V8
I2P3C+@90R4"
1040 G="R1Q7O5L8RFRERD16 V10I1P3K5E4 V4I2K10R16R1RFRERD16Q8 V10K
5E2R4"
1050 H="V14I2Q8P3K5R2.R8O2L4DC<B-A>R1DC<B-A2>R4"
1060 "!"
1070 J="L16G.F.EDC+<B>C+<AB>C+DEFGAGFEFDFA>C+D.<AB>C+DEFGAB-2"
1080 A="T160V10I1P1K5"+J+"T125V11L8Q7DB-<A>A<B->GA4"+I+"Q8A4.D2"
1090 B="V10I1P2K8"+J+"V11L8Q7DB-<A>A<B->G@27 V9I2P3A@45 V11I1P2"
+I+"@27 V9I2Q8P3A4.D2"
1100 C="V04I2P3K10R16"+J+"L8Q7DB-<A>A<B->G16 V11I1P1K5<E@60> V4I
2P3K10"+I+"16 V11I1Q8P1K5<E4.B2"
1110 D="V03I2P3K15R8"+J+"L8Q7DB-<A>A<B->G32 V9K8<E@66> V3K15"+I+
"32 V9K8Q8<E4.B2"
1120 E="R1R1R@114 V11I1Q7P1K5O6RFRERDC+4R1RFRERDQ8C+4.<I2P3F2"
1130 F="R1R1R@117 V11I1Q7P2K8O6RFRERD@27 V9I2P3C+@45 V11I1P2R1RF
RERD@27 V9I2Q8P3C+@66 <V11K5D2"
1140 G="R1R1R@126 V04I2Q7P3K10O6RFRERD16 V11I1K5E4 V4I2K11R16R1R
FRERD16 V11Q8P3K5E4.<<B2&B32"
1150 H="R1R1R@114 DC<B-A>R1DC<B-A4.G+2"
1160 "!"
1170 I="L16A8>C+8E.G.B-2A.GFEFEDC+DC<B-AGFE"
1180 A="C+2.<B4 T150V12"+I+"D8 V11>C+1&C+8"
1190 B="C+2.<B@45 V12I1P2"+I+"D@27 V9I2P3>C+1&C+@21"
1200 C="E1&E16 V4I2P3K10"+I+"D16 V11K5>E1R8."
1210 D="E1&E16. V3K15"+I+"V11K5B-1R4"
```

▶なんとSBS（静岡放送）のラジオ，毎週日曜日に「パソコンアタッククラブ」というのをやっています。4年位やっている長寿番組です。ぜひ聞こう！（まじめな番組です）
細澤　有紀（18）静岡県

```
1220 E="A1R1R1R32 V11I1P1E1&E8"
1230 F="P1<A1R1R1R16 V9K8>E1&E@21"
1240 G="V9K8<A1R1R1 V11>G1R8."
1250 H="G1R1R1R32>G1R4"
1260 "!"
1270 A="T140V10I1P1K05"
1280 B="V10I1P2K08"
1290 C="V04I2P3K10"
1300 D="V03I2P3K15"
1310 E="V10I1P1K05"
1320 F="V10I1P2K08"
1330 G="V04I2P3K10"
1340 H="V03I2P3K15"
1350 "!"
1360 J="C+EC+<B->C+<B->"
1370 K="GB-GEGE"
1380 L="EGEC+EC+"
1390 M="B->C+<B-GB-G"
1400 A="L16O6E.C+.<B->C+<B->"+J+J+J+"<"+K+K+K+K+J+J+J+J+"C+EC+"+
L+L+L+"EGE"
1410 B=A:C=A:D=A
1420 E="L16O5G.E.C+EC+"+L+L+L+"<"+M+M+M+M+L+L+L+L+"EGE"+K+K+K+K
1430 F=E:G=E:H=E
1440 "!"
1450 I=STRING$(6,"GB-")+">C+<B->"+STRING$(5,"C+E")
1460 A=I+"8.":B=A
1470 C=I+"8"
1480 D=I+"@15"
1490 H="GEGB->C+<B->C+<B->C+"+STRING$(6,"EC+")
1500 E=H+"8.":F=E
1510 G=H+"8"
1520 "!"
1530 I="V12I1P3K5"
1540 A="T70V12I2P1K5 L2O6AAGRGR1R@120"
1550 B="V09I2P3K8 L2O6AAGRGR1R@120"
1560 C="V12I2P1K5 L2O6C+D<GRER1R@120"
1570 D="V09I2P3K8 L2O6C+D<GRER1R@120"
1580 E="V12I2P1K5 L2O5EFDRC+R1R@120"
1590 F="V09I2P3K8 L2O5EFDR4P2<<A8G8>>P3C+P2L8<<EFDEC+D<B>C+<AB-G
+A4>>P3L2"
1600 G="V12I2P3K5 L2O4AA>B-RAR1R@120"
1610 H="V14I2P3K5 O3G2F2B-2.A8G8A2L8EFDEC+D<B>C+<AB-G+A4"
1620 "!"
1630 A="V13C+4D<FL1ED&D&D@99RR2"
1640 B="V10C+4D<FL1ED&D&D2RR2"
1650 C="V13E4FD1C+<L1D&D&D@99RR2"
1660 D="V10E4FD1C+<L1D&D&D2RR2"
1670 E="V13<A4AA1&A4G.F4.E4.F&F@147R1R"
1680 F="V10A4AA1&A4G.F4.E4.F&F.R1R"
1690 G="V13A4ARL1R<A&A&A@99RR2"
1700 H=">G4F2D2<L1AD&D&D@99RR2"
1710 "!"
1720 '
1730 FOR P=0 TO R-1:PLAY A(P);:NEXT:PLAY ":";
1740 FOR P=0 TO R-1:PLAY B(P);:NEXT:PLAY ":";
1750 FOR P=0 TO R-1:PLAY C(P);:NEXT:PLAY ":";
1760 FOR P=0 TO R-1:PLAY D(P);:NEXT:PLAY ":";
1770 FOR P=0 TO R-1:PLAY E(P);:NEXT:PLAY ":";
1780 FOR P=0 TO R-1:PLAY F(P);:NEXT:PLAY ":";
1790 FOR P=0 TO R-1:PLAY G(P);:NEXT:PLAY ":";
1800 FOR P=0 TO R-1:PLAY H(P);:NEXT
1810 RUN "FUGUE"
```

リスト3 トッカータとフーガ2

```
10 PLAY ""
20 DEFSTR A-N:DEFINT O-Z
30 GOTO 160
40 '
50 LABEL "!"
60 ON Z GOTO 70,80,90,100,110,120,130,140
70 PLAY A;:RETURN
80 PLAY B;:RETURN
90 PLAY C;:RETURN
100 PLAY D;:RETURN
110 PLAY E;:RETURN
120 PLAY F;:RETURN
130 PLAY G;:RETURN
140 PLAY H;:RETURN
150 '
160 FOR Z=1 TO 8
170 A="T90V9I1Q8P1K05" :E=A
180 B="V9I1Q8P2K08 R64":F=B
190 C="V4I2Q8P3K10 R16":G=C
200 D="V3I2Q8P3K15 R08":H=D
210 "!"
220 I="L16RAGAFAEADAC+ADAEAFA<A>A<B>AC+ADAC+ADAEA"
230 A="R1R1O6L16RDCD<B->D<A>D<G>D<F+>D<G>D<A>D<B->D<D>D<E>D<F+>D
<G>D<F+>D<G>D<A>D"
240 B=A:C=A:D=A
250 E="O5"+I+"L8FF+GC<B-AB->CD<F+GAB-AB-F+"
260 F=E:G=E:H=E
270 "!"
280 A="Q5L8<B->D<B->DE-<G>E-<G>C<A>C<A>D<F>D<FB-GB-G>C+<E>C+<EAF
AFGC+GC+FDFDE<B->E<B-"
290 B=A:C=A:D=A
300 G="L16G>G<G>GDGDG"+STRING$(4,"CE-")+STRING$(4,"CF")+STRING$(
4,"<B->D")+STRING$(4,"<B->E")+STRING$(4,"<A>C+")+STRING$(4,"<F>F
")+"<"+STRING$(4,"EB-")+STRING$(4,"DA")+STRING$(3,"EG")+"E"
310 E=G+"G"
320 F=E+"@15"
330 H=G+"32"
340 "!"
350 A="Q8O6"+I+"FAEADACA<B->ACADG<B->GEGDGCG<B->G<A>G<B->GCF<A>F
DFCF<B->F<A>F<G>F<A>F<B->E<G>EC+E<B->E<A>E<G>E<F>E<G>E<A>D<F>D"
360 B=A:C=A:D=A
370 I="L8O4"+STRING$(16,"A")+"L4A>DDRCCCR<B-B-B-RAAAL8R>D"
380 E="V9I1P1K5"+I
390 F="V7I2P3K8"+I
400 I="L8O4FEDGFEFC+DC+DEFEFC+L4DFGRCEFR<B->DER<A>C+DR8F8"
410 G="V9I1P1K5"+I
420 H="V7I2P3K8"+I
430 "!"
440 D="<E>E<E>E<F>D<F>D<B->C+<B->C+<A>D<F>D<E>E<E>E<F>D<F>DRDC+D
<B>DC+<BRAGAEGFE>RDC+DFD"
450 A=D+"C+<B"
460 B=A:C=A
470 I="L8C+C+DDE4RDC+C+DDL4ER>C+R<FDR"
480 E=I+"64"
490 F=I+"32"
500 G="L8B-B-AAG4RAB-B-AAG+4R4A4R2."
510 H=G
520 "!"
530 A="V9I1P1K5":D=A:G=A
540 B="V9I1P2K8":E=B
550 C="V5I2P3K10":F=C
560 H="V7I2P3K8"
570 "!"
580 I="L4O6C+E.DC+8C<B-AAGGF+8A>E-8D"
590 A=I+"R8GF+8GL8<B->DDDDDDDDDDL16DGFG"
600 B=A
610 C=I+"V9I1P1K5R8.<D4L8RDDDDC<B->DDC<B->DDC<B-4"
620 I="L4O5R2FED8AG.F+FE-.D8F+8>C<B-16A16B-8"
630 D=I+"B-8AL8>DC<B-AB-F+GF+GAB-AB-F+L16GB-AB-"
640 E=D
650 F=I+"B-@15 V9I1P2K8D4L8RDDDDC<B->DDC<B->DDC<B-4"
660 G="L16O4A>AGAC+GFERDC+D<AAGAF+>DC+D<G>FE-DC+E<A>C+<D>E-DC<B>
D<GBC>DC<B-A>C<F+AD>C<B-AB->AGF+G<B-AGRDCD<B->D<A>D<G>D<F+>D<G>D
<A>D<B->D<D>D<E>D<F+>D<G>D<F+>D<G>D<A>D<B-4"
670 H=G
680 "!"
690 I="EFDECAGAFGEFDB-AB-GAFGE>C<B->C<AB-GAFE-DCDC<B-AB->D<B-AGB
-GFEFGAB->DC<"
700 A=I+"B-A4"
710 B=A
720 C="V5I2P3K10 L16R"+I+"V9I1P1K5F4R16"
730 I="GAFGE>C<B->C<AB-GAF>DCD<B->C<AB-G>EDECD<B->C<A8F8B-AGFGB-
GFEGEDCDEFG8E"
740 D=I+"8L16RC<B->CR8"
750 E=I+"8L16RC<B->C32."
760 F="V5I2P3K10 L16R@9"+I+"@15 V9I1P2K8F4"
770 G="L8RB>C4RC+D4RDE4REF<AB->D<G4RB-4AG>C<F4R16"
780 H=G+"."
790 "!"
800 A="V9I1P1K05":E=A
810 B="V9I1P2K08":F=B
820 C="V4I2P3K10":G=C
830 D="V3I2P3K15":H=D
840 "!"
850 A="L8O6C<B-AGAB->C<EFGAGAB-L16>C<B-AGFE-DC<B-8>>R4.EDC<B-AGF
ED8>R4.GFEDC<B-AG"
860 B=A:C=A:D=A
870 E="L16O4A>C<G>C<F>C<E>C<F>C<G>C<A>C<C>C<D>C<E>C<F>C<E>C<F>C<
G>C<A4R4>>DC<B-AGFEDC8>R4.FEDC<B-AGFE8R4."
880 F=E:G=E:H=F
890 "!"
900 I=STRING$(2,"AFEFCFEF")+STRING$(2,"GEDECEDE")
910 A=">"+I+"_5"+I+"FGFEDC<BA"
920 B=A:C=A:D=A
930 I="FR>CR<FR>CR<ER>CR<ER>CR<"
940 E="L8"+I+"_5"+I+"DR4."
950 F=E:G=E:H=E
960 "!"
970 I="BGB>DFAFD<"
980 J="B-GB->CEGEC<"
990 K="AFA>CDFD<B-"
1000 L="GEGB->C+EC+<B-"
1010 A="¯5"+I+"_"+I+"¯"+J+"_"+J+"¯"+K+"_"+K+"¯"+L+"_"+L
1020 B=A:C=A:D=A
1030 E="¯5_R1R1R1R1"
1040 F=E:G=E:H=E
1050 "!"
1060 I=">RAGAFAEADAC+ADAEAFA<A>A<B>AC+AFA"
1070 A=I+"EAL8FEDC<B-AL16B-AGFEDC+<B"
1080 B=A:C=A
1090 D=I+"V9_I1P1K5<RAGARF+EF+RDCDG8 V3_I2P3K15R4EDC+<B"
1100 I="A>C+D<GFAB>C+DC+DEDC+"
1110 E=I+"DC+D<AGF+L16<B->C<B-AGFED"
1120 F=E:G=E
1130 H=I+"64 V9_I1P2K8L16<RAGARF+EF+RDCDG8 V3_I2P3K15R@45<GFED"
1140 "!"
1150 I="C+<A>C+EGB-GE"
1160 J="D<A>DFA>D<AF"
1170 K="EC+EGB->C+<B-G"
1180 L="FDFA>DFD<A"
1190 M="GEGB->C+EC+<B-"
1200 A="AR@84L1RRRRRRRR"
1210 B=A:C=A:D=A
1220 E=STRING$(2,"¯"+I+"_"+I+"¯"+J+"_"+J)+STRING$(2,"¯"+K+"_"+K+
"¯"+L+"_"+L)+"¯"+M+"_"+M+"¯AFA>C+DFD<A"
1230 F=E:G=E:H=E
```


▶この前発見したんですが，メタルサイトをクリアしてFinという文字が出てから数分後。ディスクが出てきて，ディスク交換の指定が表示され，入れ替えると，なんと「メリークリスマス」！

星野 大 (17) 新潟県

```
1240 "!"
1250 A="RRR4¯L16>RFEDAR4>EDC+DL32E-DC<B-AG"
1260 B=A:C=A:D=A
1270 E="B->D<B-GFAFD<A>D<AFD>DC+<B>C+B-AGFGFEDB-AGFGFEDL32EFGAB>
C+D16R8.<A16B>C+DEFGL16AGFEF8R8"
1280 F=E:G=E:H=E
1290 "!"
1300 A="L8F+A4G>C<B>E-DE-<B>C<B>CDE-DE-FL32"+STRING$(33,"GF")+"L
16GFGE-FDE-CFE-FDE-CD<B->E-DE-CD<B->C<A>DCD<B->C<AB-GB-AB->C<B-A
G"
1310 B=A:C=A
1320 E="RE-DC<B-8B8>C8G8L32"+STRING$(25,"GF")+"L16GFGE-GDGCG<B>G
CGDGE-G<G>G<A>G<B>GCG<B>GCGDGE-8D8GA-FGE-8C8FGE-FD8<B-8>E-FDE-C8
F+8GAFGE-GF+GAGF+E"
1330 F=E:G=E
1340 I="L16O3RGFGE-GDGCG<B>GCGDGE-G<G>G<A>G<B>GCG<B>GCGDGL8E-DC<
B>CDE-FG<B>CDE-DE-FG<B>CDE-<AB->CD<GAB->CDGDE-<B-A>C"
1350 D="V10I2P1K5R8"+I
1360 H="V10I2P2K8R@27"+I
1370 "!"
1380 A="F+8A8>D<G>C<F+B-G>D<AB-GAF+G>D<F+>D<G>D<A>D<B-G>D<AB-G>C
<A>DC<B-AB-AB-GAB->CDE-DCD<B->C<AB-G8B8>CE-FGA-GFGE-FDE-"
1390 B=A:C=A
1400 E="RDCD<B->D<A>D<G>D<F+>D<G>D<A>D<B->D<D>D<E>D<F+>D<G>D<F+>
D<G>D<A>D<B-8A8G2F+4G8A8B-AGFE-8>E-8D4C8D8"
1410 F=E:G=E
1420 D="D1&D1&D4R<G>C<A>D<DG4RG>C4.<B>C4"
1430 H=D
1440 "!"
1450 C="C8<B-8A8B-4A4G4F+8F8E-4DB-AB-AGF+>E-DC<B->AGF+G8.FE-FD"
1460 A=C+"E-"
1470 B=A
1480 D="V3I2P3K15R8L16O6"+C+"64"
1490 G="E-FDE-CE-DFGFE-FD<B->E-DC+<A>DC<BG>C<B-AFB->DCDC<B-A>C<B
-AG>C<B-AB->C<B-AGAF"
1500 E=G+"G"
1510 F=E
1520 H="V3I2P3K15R@21L16O5"+G+"64"
1530 "!"
1540 A="V9I2P1K5":C=A:E=A:G=A
1550 B="V9I2P2K8":D=B:F=B:H=B
1560 "!"
1570 A="O6L4C+DC<B-"
1580 B=A
1590 I="O5L16RB-AGA4F+4GFE-D"
1600 C=I+"R"
1610 D=I+"32."
1620 I="O5L16R4RE-DCDC<B-AG4R"
1630 E=I+"64"
1640 F=I+"@9"
1650 G="O4L4EF+DG"
1660 H=G
1670 "!"
1680 A="V9I1P1K5":D=A
1690 B="V9I1P2K8":E=B
1700 C="V5I2P3K10":F=C
1710 G="V10I1P1K5"
1720 H="V8I2P3K8"
1730 "!"
1740 A="L8O5A>C+DEF4E2D4.FGDC+DEDC+DEDC+DEGFD<B->EDDC+4DEFEFC+DC
+DEFEFGL16AB-GAFGEFD<A>C+<A>D<A>E<A>F<A>E<A>F<A>G<A>"
1750 B=A:C=A
1760 D="O5C+8E8F8G8<D8>>D4.C+4L16<AB-GAL8FAGAGFGFEAB-AGFA>ED<AGG
AFL16EAGAFAEADAC+ADAEAFA<A>A<B>AC+ADAC+ADAEAL8FEDGFEFC+DC+D<B>"
1770 E=D:F=D
1780 G="L16O4AB-AGFGEA>AGAFGAGAB-AGAFGEFDC<B-AB-AGFEFEDC+<A>D<A>
E<A>F<A>G<A>F<A>E<A>D<A>C+A<A>ADAF>D<G>D<E>C+<F>D<D>D<L1A2&A&A&A
&"
1790 H=G
1800 "!"
1810 A="A<A>E<A>F<A>G<A>A4R1R2.REDC+D8RFEFGAB-AGAF8E8DEFE-DC<B-A
GAB->C<A4R8>C4<B-8R8>G8E-FDE-C<B-AGA4"
1820 B=A:C=A
1830 I="C+GFDL16C+<<¯AGAFAEADAC+ADAEAFA<A>A<B>AC+ADAC+ADAEA_>>RG
FEFC+D4.C+4DAB>C+D8C8<F4.E-8.C<AFA>CE-CD<B-GDGB->DGCD<B->C<AGF+E
F+DF+G"
1840 D=I+"R8"
1850 E=I+"@9"
1860 F=I+"64"
1870 I="A&A&A2<¯L8F4RFGEA<A>D4R<AB-G>C<CF4RAB-4RB->C4<RA>D4R"
1880 G=I+"16"
1890 H=I+"@21"
1900 "!"
1910 A="V9I1P1K05":E=A
1920 B="V9I1P2K08":F=B
1930 C="V4I2P3K10":G=C
1940 D="V3I2P3K15":H=D
1950 "!"
1960 C="L16O4A>D<AG_F+DF+GA>D<AF+¯GDGAB->D<B-A_GDGAB->DC<B-¯AF+A
B->CE-C<B-_AF+AB->CE-C<B-¯AB->DF+GB-GD<_B-GB->DGB-GDC<A>CE-F+AF+
E-C<A>CE-F+AF+"
1970 A=C+"E-"
1980 B=A
1990 D=C+"64"
2000 E="L4O3RDR"+STRING$(4,"¯DR_DR")
2010 F=E
2020 G=E+"8."
2030 H=E+"@27"
2040 "!"
2050 A="V10I2P1K5":C=A:E=A
2060 B="V10I2P2K8":D=B:F=B
2070 G="V12I2P1K5"
2080 H="V12I2P2K8"
2090 "!"
2100 A="L8O6R4DC<B-4>CDE-4FGA4FE-"
2110 B=A
2120 C="L8O5DRGF+G4GB>C4CEF4<B-A"
2130 D=C
2140 E="L8O4F+RGAB-4>E-<BG4>C<B-A4B->C"
2150 F=E
2160 G="L16O3RDCD<B->D<A>D<G>GFGE-GDGCC<B->C<A>C<G>C<F>FE-FDFCF"
2170 H=G
2180 "!"
2190 I="E-DC<B->C<B-AG>C<B-AGAGF+GFEDC+8E8FGAB>C+DEFG8FEL8DC+DEF
GAC+DEFEFD"
2200 A="L16D"+I
2210 B=A
2220 E="B-8 V5I2P3K10L16>"+I+"16"
2230 I="FE-DE-DC<B->E-DC<B->C<B-AB-AGFE8G8AB>C+DEFGAB-8AGFAEADAC
+ADAEAFA<A>A<B>AC+ADAC+ADA<B>"
2240 C="L16DG"+I+"A"
2250 D=C
2260 F="B-8 V5I2P3K10R@9L16"+I
2270 G="L8<B->E-F<FG>CD<DG4R16A16G16A16FDA4D1&D2>R4.D
2280 H=G
2290 "!"
2300 I="T70EDDC+T55DFE4.D"
2310 A=I+"2.R4."
2320 B=I+"@141R4."
2330 I="L8C+DGEFAB-.A.<F"
2340 C=I+"2.R4."
2350 D=I+"@141R4."
2360 E="V10P1K5L8<RAB-AA>DD.C.<F2.R4."
2370 F="V10P2K8R64L8RAB-AA>DD.C.<F@141R4."
2380 I="<AFGAFDG.A.B-"
2390 G=I+"2.R4."
2400 H=I+"@141R4."
2410 "!"
2420 A="T150V8I1P1K05":E=A
2430 B="V8I1P2K08 R64":F=B
2440 C="V4I2P3K10 R16":G=C
2450 D="V3I2P3K15 R08":H=D
2460 "!"
2470 A="L16O6F8GAB-<AB->CDCDE-F<FGAB-AB->CD<AGFE-GAB->C<GFE-DFGA
B-<AB->CDCDE-F<FGAB-AB->CDC<B-AGB->CDE-DC<B-A>CDEF+E-DC<B->F+T10
0GAB->C"
2480 B=A:C=A:D=A
2490 E="R1R1R1R1R@168"
2500 F=E:G=E:H=E
2510 "!"
2520 B="D.<G.> T70V10I2E2.D2.R1R1 T80V11D4C1&C1"
2530 A=B+"&C64"
2540 C="D.<G32 V10P1K5G2.F4D2R1R1 V11G+4A2.G8F8E1&E64"
2550 D="D@15 V10P2K8<G2.F4D2O3L8R4G+.BFG+DF<B>D<G+A4R@60 V11O5G+
4A2.G8F8<E1"
2560 F="O4G8. V10I2>C+2.D4<B2R1R1 V11G+4A1&G1"
2570 E=F+"&G64"
2580 G="O4G8 V10K5>E2.A4F2R1R1 V11<B4>C1&C1&C64"
2590 H="O4G16 V11K5<C+2F4.D8<G+2 T100L8>R4G+.BFG+DF<B>D<G+A4 V13
E1&E1&E@111"
2600 "!"
2610 A="T150V8I1P1K05"
2620 B="V8I1P2K08 R64"
2630 C="V4I2P3K10 R16"
2640 D="V3I2P3K15 R08"
2650 E="":F=E:G=E:H=E
2660 "!"
2670 A="R1L16O5G.FGEEDECAGAFFEFDBABGGFGE>C<B>C<A>DCD<B>EDECFEFDG
FGEC<B>C<A>DCD<BG+F+G+E>C<B>C<AFEFDBABCAGA<B>G+F+G+EC<BAG+>DC<BA
>EDC<B>FEDCGFEDAGFE8¯G8A8B8"
2680 B=A:C=A:D=A
2690 E="L1RRRRRRRR@138"
2700 F=E:G=E:H=E
2710 "!"
2720 A="T120V11I2P1K5 O6C1&C8"
2730 B="V11I2P2K8 O6C1&C8"
2740 C="V11I2P1K5 R@108O5E2"
2750 D="V11I2P2K8 R@99O5E2"
2760 E="V11I2P1K5 R@120O4G2"
2770 F="V11I2P2K8 R@123O4G2"
2780 G="V11I2P3K5 R@120O5G2"
2790 H="V12I2P3K5 R@120O3E2"
2800 "!"
2810 A="L16C+2.R4 T145<<A>C+EA>L8C+RC+RDRL16<<A>DFA>L8DRDR<BRL16
<GB>DGL8BRBR>CRL16<<G>CEG>L8CRCR<ARL16<FA>CFL8ARARB-RL16<FB->DFL
8B-RB-RB-RL16<EGB->C+L8B-RB-RARL16<DA>DF+A8R8"
2820 B=A
2830 C="E2.R4Q4L4R2EDR2FDR2DER2ECR2CDR2DER2C+DR2Q8"
2840 D=C
2850 E="A2.R4Q4L4R2AAR2AGR2GGR2GAR2FFR2FER2EF+R2Q8"
2860 F=E
2870 G="L16A2.R4RV5K10<A>C+EA>C+8RV11K5<A8R8F8R8.V5K10<A>DFA>D8R
V11K5<A8R8G8R8.V5K10<GB>DGB8RV11K5G8R8G8R8.V5K10<G>CEG>C8RV11K5<
G8R8F8R8.V5K10<FA>CFA8RV11K5F8R8F8R8.V5K10<FB->DFB-8RV11K5F8R8G8
R8.V5K10<EGB->C+B-8RV11K5G8R8F+8R8.V5K10<DA>DF+A8RV11K5"
2880 H="G2.R4Q4L4R2GFR2FFR2FER2EER2EDR2DC+R2C+CR2Q8"
2890 "!"
2900 B="T100L1A2.GF2ET80D&D&D&D2."
2910 A=B+"&D64"
2920 D="L1D4<B->E2DC<B-2.A&A&A2"
2930 C=D+"&A64"
2940 F="L1F+4GB-2A2AF2G2.<F&F&F2"
2950 E=F+"&F64"
2960 G="L1F+4DC+2RRR2.<<D&D&D@99"
2970 H="L2C4<GB-EFA1B-G.D1&D1&D@99"
2980 "!"
2990 '
3000 PLAY ":";
3010 NEXT
3020 PLAY ""
```

▶あの事件以来ウイルスの怖さがわかった。このままでは人間不信（機械不信？）になりそう。

大内　善雅（18）福島県

★(で)のショートプロぱーてぃ その11

前夜祭はすたっきぃだ!

Komura Satoshi 古村 聡

やったー，今月は1周年だ。と思ったら勘違いだって……，とほほほ。まあいいか楽しみは先にとっておきましょう。そういうわけで今月は楽しい画面消去プログラム「ぱっくりあ」とパズルゲーム「STACKY」の2本です。

illustration : T.Takahashi

どもー。のーみそが1フレーズ足りないといわれる（で）です。

しょえーい，なんでこんなに凡ミスばっかりするんだよぉ，わしって！ そーです，読者の方からご指摘があったようにあのゼッパチマンの歌は1フレーズ書き落としていたのです。ああ，それだけならまだしも先月,「次はショートプロぱーてぃ1周年記念ぱーてぃ」だとかいってたけど，よく考えてみたら1周年は来月じゃないか！……だーっ，凡ミス大爆発！ 1日や2日，日にちを間違えるのはまだしも（普通はそれだってしないと思う）1カ月単位で間違えるなよー，何をやってんだかな本当に。

まあしょうがない，こうなったら今月は「次はショートプロぱーてぃ1周年記念ぱーてぃだから，今夜はショートプロぱーてぃ1周年記念ぱーてぃ前夜祭ぱーてぃ」を行うぞ，だぜ！（冗談です，本当にごめんなさい。ささやかなショートプロ1周年記念ぱーてぃは来月やります。しくしく。まあ，でも締め切りが1カ月延びたと思えばね……）

ぱっくぱっくでCLS!

いやー，また今月もX68000のショートプロはマシン語だなー。ま，今月はマシン語特集だそうだし，いいか。

では，今月の1本目。今月の1本目は5月号掲載sp_chk( )の豊田さんで，ふつーのCLSじゃつまんない！ そんなふつーでないあなたのカスタムCLSプログラム,「ぱっくりあ.X」です。

ぱっくりあ.X for X68000
(command.x用)
神奈川県 豊田明彦

どういうものかっていうと早い話が画面をクリアするプログラムなんですが，ただ画面をクリアするだけじゃあつまんない！んで，画面の文字をパックマンみたいなアニメーションがぱくぱくと消してってくれたりするんだな，これが。ちょっとアニメーションのパターンのドットが粗いような気もするんだけど（16×16ドットのキャラクターを無理やり32×32に伸ばして使ってるんだそーな)。ま，細かいことは気にしない気にしない。

で，例によって打ち込み方。今月はマシン語特集でそっちのほうでも書いてるんですけど（というより，書く予定なんだ，これから。あはは，せっかく提出した特集の原稿，総ボツくらってこれからぜーんぶ書き直しなんだよぉ……しくしく。ライターってつらい……)。とりあえずざっと説明しちゃいましょう。まず，

1) 先月号のディスクのおまけの
IOCSCALL.MAC
DOSCALL.MAC
をアセンブラ用に使っているディスクにコピーする。
2) ED.Xを立ち上げる（コマンドラインから ED pacls.sと打つ)。
3) 載っているリスト打ち込む！
4) ESC [E] でED.Xを終わる。
5) AS PACLS と打ってアセンブル。
6) No Fatal error(s) と出なかったら2)に戻って間違えてるところを直す。
7) LK PACLS と打ってリンクすれば，……いっちょあがり！ です。

プログラムリストをじーっとながめればわかるけど，これって簡単にアニメパターンを描きかえられるうえに，アニメパターンが通ったあとの忘れ物まで描ける！ つまりちょいちょいっと手を加えれば「泥棒がものを落としながら走っていくCLS」なんてーのも簡単にできるんだな。ほら，最後のほうになんか0と1で絵が描いてある部分があるでしょ，そこをちょいちょいと手直ししてさっきの打ち込みと同じ手順でアセンブルしてやるだけなんですねー。投稿原稿にもせっかく「アニメパターンを4倍角にしたおかげでエディタで簡単にアニメパターンが描けますので気に入らない方は各自描き直してください」ってあることだし，ここは存分に描き直させていただきましょう。私もおもわずイースIIIのクモ君にしてみたりしました（気持ち悪かった……)。

そうそう，投稿原稿で思い出したんですが「sp_chk( )同様，このプログラムもフリーウェアにしたいと思います。ネットに流そうが改造しようが煮るなり焼くなりお好きにどうぞ」って書いてあったんですよねー。ディスク集に間に合わなかったのがとても残念。電脳倶楽部にでも投稿しちゃおっか？

ワンパをねらえ！ スタッキーだ！

さて今月2本目のプログラムは2月号に掲載されたNUMBERSの坂本さんの投稿で，投稿原稿いわく（坂本さんの投稿原稿って好きなんだよねー，なんか楽しくって）「前作NUMBERSに続いてまたもやX1のグラフィック機能をまったく無視した低解像度ショートパズルゲーム第2弾「STACKY」（略称勝手に成りジャン，ぢゃなくてスタッキー)」だそうです。あぁ，つ，つかれる……。

STACKY for X1シリーズ
(CZ-8FB01用)
秋田県 坂本康

◎遊び方

ばらばらに並んでいるブロックがありますね。こいつを横1列に同じものが並ぶように並びかえるインテリジェンスたーっぷりのパズルゲーム（ひょえーい）だったりします。

こんなふうにね。

（ちょっとブロックの形が違うけど）

ただし一度に動かせるのは1つだけで上から順番に動かさなくてはいけません。テンキーの4，6で矢印をみぎひだりに動かして選びスペースキーで決定します。ESCキーでメニュー画面になります。

そうそう，メニュー画面では，

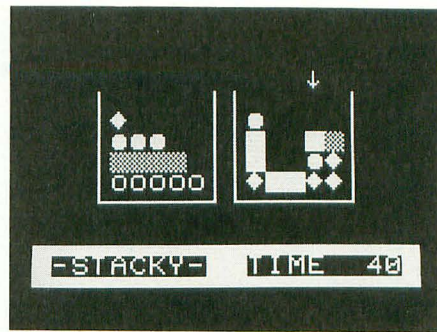
TRY AGAIN……途中で詰まったときのやりなおし。

CORRECT ANSWER……どうしても解けないとき，解答の手順を見せてくれる。

MODE CHANGE……ブロック数を3×3から8×8の範囲で選べてスペースキーで決定。

ということがそれぞれできるようになってます。

ようし，積んで積んで，また積んで……。あ，そっか，積んでいく様子がスタックを思わせるからスタッキーなわけか。納得。しかし，なんですね，「解答モード付きパズルゲーム」っていうのはいいもんですねー。自分が解けなかったらコンピュータがやってるとこながめてりゃいいんだから。コンピュータのほうが人間さまより頭がいいというのもなんとなくくやしいけど……。いいか，どうせ1フレーズ抜けた男だもん(いじけきってます，はっきりいって)。

さて，今月はいままでショートプロに登場したおふたりさんの登場だったわけですが，そろそろ常連さんっていう人達も出てきそうですねー。なんか，私はまだよく見てないんだけど，投稿整理係さんの話だと「夜，見ないよーに」のデモの太田さんとかの新しい投稿とかももう届いてるって話だし。はたして誰が常連第1号と認定されるのか？

STACKY

それにしても，やっぱり，得意分野があるのって強いですよねー。とくにスタッキーの場合もあの4倍角文字のやつを作った人のですよっていっただけで，編集室でもなるほどってわかる人が多かったくらいですからねー。世の中ワンパターンというとなにか悪いことのように聞こえるけど，私はそんなことって絶対ないと思います。だって，一流とか有名なブランドって必ずワンパターンじゃないですか。やっぱりワンパって人によく憶えてもらえるし，人にアピールするときに「私は○○で有名な××（名前）だ！」っていえるのって凄く強いと思うんですよね。それに特にプログラムを組む場合ってジャンルや言語を同じものでやってると技術の蓄積っていうのがありますから，上達が早いんですよね。

さあ，そういうことでみんなでワンパを目指せ！ 常連への道は近いぞ！ というわけでまた来月。

リスト1 ぱっくりあ

```
 1:          .list
 2: *
 3: *ぱっくりあ. X
 4: *              By Toyo
 5: *
 6: *               1990/05/14
 7:          .include DOSCALL.MAC                    *IOCS・DOSコール使用
 8:          .include IOCSCALL.MAC
 9:          .text
10: _txtvrm           equ      $e00000               *テキストVRAMの先頭
11:
12: putank:           lea      usersp,sp             *スタックの設定
13: main:             clr.b    d3                    *アニメパターンのカウンタ
14:                   clr.b    d1                    *Y方向のカウンタ
15: lopmn2:           clr.b    d0                    *X方向のカウンタ
16: lopmn:            addq.b     #1,d3
17:                   cmp.b      #4,d3
18:                   bcc      prnl
19: prn0:             lea      fntdat,a0             *パターン1処理
20:                   bsr      loopnai
21:                   bra      nxtmn
22: prn1:             lea      fntdat2,a0            *パターン2処理
23:                   bsr      loopnai
24:                   cmp.b    #8,d3
25:                   bne      nxtmn
26:                   clr.b    d3                    *2の表示後，1の表示
27: nxtmn:            addq.b   #1,d0                 *"next x"
28:                   cmp.b    #24,d0
29:                   bne      lopmn
30:                   addq.b   #1,d1
31:                   cmp.b    #2,d1
32:                   bne      lopmn2
33: cls:              move.w   #2,-(sp)              *DOS上のCLS処理
34:                   move.w   #10,-(sp)
```

```
35:                   DOS      _CONCTRL
36:                   addq.l   #4,sp
37:                   move.w   #0,-(sp)
38:                   move.w   #14,-(sp)
39:                   DOS      _CONCTRL
40:                   addq.l   #4,sp
41:                   DOS      _EXIT                 *作業終了
42:
43: loopnai:          bsr      putptn                *上ループ内をサブルーチン化
44:                   bsr      wait
45:                   bsr      clrptn
46:                   rts
47: putptn:           move.b   #8,d4                 *パターンのPUT
48: mlop01:           bsr      putpl                 *7回分まとめてサブルーチン化
49:                   add.b    #2,d1
50:                   subq.b   #1,d4
51:                   bne      mlop01
52:                   sub.b    #16,d1
53:                   rts
54: clrptn:           lea      fntdat3,a0            *PUTを使って画面クリアも
55:                   bsr      putptn
56:                   rts
57:          *パターンの出力
58: putpl:            movem.l d0-d6/a0-a6,-(sp)      *細かいサブルーチン
59:                   bsr      prn32
60:                   movem.l (sp)+,d0-d6/a0-a6
61:                   rts
62:          *wait:その名の通りウェイト
63:          *
64: wait:             movem.l d0-d6/a0-a6,-(sp)
65:                   move.l   #100000,d5            *wait counter=140000
66: waitl:            nop
67:                   dbra     d5,waitl
68:                   movem.l (sp)+,d0-d6/a0-a6
```

```
 69:             rts
 70:         *prn32:16bitのｷｬﾗｸﾀｰを32bitに拡張して
 71:         *       32bit座標の(X、Y)に表示する
 72:         *in:    d0.b = X
 73:         *       d1.b = Y
 74:         *       a0   = 16ﾋﾞｯﾄﾌｫﾝﾄの位置
 75: prn32:
 76:                 lea     extfnt,a1
 77:                 bsr     ex1632
 78:                 bsr     xyloc32
 79:                 bsr     put32
 80:                 add.l   #$20000,a0
 81:                 bsr     put32
 82:                 rts
 83:
 84:         *xyloc32:XY座標→VRAMｱﾄﾞﾚｽ
 85:         *in:    d0.b = X
 86:         *       d1.b = y
 87:         *out:   (a0) = VRAM Address
 88: xyloc32:        movem.l d0-d1,-(sp)             *待避
 89:                 and.l   #$000000ff,d0
 90:                 and.l   #$000000ff,d1
 91:                 lsl.l   #2,d0
 92:                 lsl.l   #2,d1
 93:                 lsl.l   #7,d1
 94:                 lsl.l   #3,d1
 95:                 add.l   d1,d0                   *d0=y*$80+x
 96:                 move.l  #_txtvrm,a0
 97:                 add.l    d0,a0                  *a0=d0+a0
 98:                 movem.l (sp)+,d0-d1             *復帰
 99:                 rts
100:
101:         *put32: 32dotsでｷｬﾗ表示
102:         *in:    (a0) = 表示address
103:         *       (a1) = ﾌｫﾝﾄの格納位置
104:         *out  : none
105: put32:          movem.l d0-d1/a0-a2,-(sp)  *待避
106:                 move.l  a1,a2
107: SP1:            clr.l   a1                      *SUPERへ
108:                 IOCS    _B_SUPER
109:                 move.l  d0,a6
110:                 move.b  #32,d0
111: ldir1:          move.l  (a2)+,(a0)
112:                 lea     $80(a0),a0
113:                 subq.b  #1,d0
114:                 bne     ldir1
115: US1:            move.l  a6,a1                   *ﾕｰｻﾞｰﾓｰﾄﾞへ
116:                 IOCS    _B_SUPER
117:                 movem.l (sp)+,d0-d1/a0-a2  *復帰
118:                 rts
119:         *
120:         *ex1632 :16*16ﾄﾞｯﾄﾌｫﾝﾄと32*32に拡張
121:         *in     :a0  16ﾄﾞｯﾄﾌｫﾝﾄの位置
122:         *       :a1  拡張ﾌｫﾝﾄの格納位置の先頭番地
123:         *out    :(a1)拡張ﾌｫﾝﾄ
124:         *break  :a0(次のﾌｫﾝﾄの位置)
125: ex1632:
126:                 movem.l d0-d1/a1,-(sp)
127:                 move.b  #16,d1
128: exllop:         move.w  (a0)+,d0
129:                 bsr     bitext
130:                 move.l  d0,(a1)+
131:                 move.l  d0,(a1)+
132:                 subq    #1,d1
133:                 bne     exllop
134:                 movem.l (sp)+,d0-d1/a1
135:                 rts
136:         *
137:         *BITEXT :ﾋﾞｯﾄ幅を倍にする
138:         *IN     :d0.w
139:         *OUT    :d0.l
140:         *BREAK  :none
141: bitext:         movem.l  d1-d3,-(sp)
142:                 clr.l   d2                      *d2には拡張した数字
143:                 move.l  #15,d1                  *調べるﾋﾞｯﾄ
144: belopl:         rol.w   #1,d0
145:                 rol.l   #2,d2
146:                 move.b  d0,d3
147:                 and.b   #1,d3
148:                 beq     benxt1                  *0ならとばす
149:                 or.l    #%11,d2
150: benxt1:         dbra    d1,belopl
151:                 move.l  d2,d0
152:                 movem.l (sp)+,d1-d3
153:                 rts
154:                 .data
155:                 .even
156: *                        0         1     1   ←何ﾋﾞｯﾄ目か
157: *                        0123456789012345
158: fntdat:         .dc.w   %0000011111100000  *ｱﾆﾒﾊﾟﾀｰﾝ1
159:                 .dc.w   %0011111111111100
160:                 .dc.w   %0111111111111110
161:                 .dc.w   %0111111111111110
162:                 .dc.w   %1111111111111000
163:                 .dc.w   %1111111111100000
164:                 .dc.w   %1111111110000000
165:                 .dc.w   %1111111000000000
166:                 .dc.w   %1111110000000000
167:                 .dc.w   %1111111100000000
168:                 .dc.w   %1111111110000000
169:                 .dc.w   %1111111111110000
170:                 .dc.w   %0111111111111100
171:                 .dc.w   %0111111111111110
172:                 .dc.w   %0011111111111100
173:                 .dc.w   %0000011111100000
174:
175: fntdat2:        .dc.w   %0000011111100000  *ｱﾆﾒﾊﾟﾀｰﾝ2
176:                 .dc.w   %0011111111111100
177:                 .dc.w   %0111111111111110
178:                 .dc.w   %0111111111111110
179:                 .dc.w   %1111111111111111
180:                 .dc.w   %1111111111111111
181:                 .dc.w   %1111111111111111
182:                 .dc.w   %1111111111111111
183:                 .dc.w   %1111111111111111
184:                 .dc.w   %1111111111111111
185:                 .dc.w   %1111111111111111
186:                 .dc.w   %1111111111111111
187:                 .dc.w   %0111111111111110
188:                 .dc.w   %0111111111111110
189:                 .dc.w   %0011111111111100
190:                 .dc.w   %0000011111100000
191:
192: fntdat3:        .dc.w   %0000000000000000  *画面上に残るもの
193:                 .dc.w   %0000000000000000
194:                 .dc.w   %0000000000000000
195:                 .dc.w   %0000000000000000
196:                 .dc.w   %0000000000000000
197:                 .dc.w   %0000000000000000
198:                 .dc.w   %0000000000000000
199:                 .dc.w   %0000000000000000
200:                 .dc.w   %0000000000000000
201:                 .dc.w   %0000000000000000
202:                 .dc.w   %0000000000000000
203:                 .dc.w   %0000000000000000
204:                 .dc.w   %0000000000000000
205:                 .dc.w   %0000000000000000
206:                 .dc.w   %0000000000000000
207:                 .dc.w   %0000000000000000
208: extfnt:         .ds.l     32
209:                 .ds.l     1024                  *ｽﾀｯｸ
210: usersp:
211:                 .end
```

リスト2 STACKY

```
1 '-STACKY-  Version 1.6  '90/4/1
2 '         Program Arranged by Y.Sakamoto
10 WIDTH 40:SCREEN:CSIZE3:DEFINT A-Z:C$=CHR$(31):E$=CHR$(27)
20 DIM B(7,7),C(7,7),P(8),T(7),W(80):M=4
30 N=9-M:U=14-2*M
40 FOR J=0 TO M:FOR I=0 TO M:C(I,J)=-1:NEXT:T(J)=J:NEXT
50 FOR I=0 TO M:SWAP T(I),T(INT(RND*(M+1))):NEXT
60 FOR J=0 TO M:FOR I=0 TO M:B(I,J)=T(J):NEXT:NEXT
70 FOR K=0 TO (M+1)^2-1
80 I=INT(RND*(M+1)):J=0
90 IF B(I,J)=-1 THEN J=J+1:IF J>M THEN 80 ELSE 90
100 X=INT(RND*(M+1)):Y=M
110 IF C(X,Y)>-1 THEN Y=Y-1:IF Y<0 THEN 100 ELSE 110
120 C(X,Y)=B(I,J):B(I,J)=-1:W(K)=X:NEXT
130 '
140 CLS:LINE(0,20)-(39,23),"■",BF:LOCATE0,20:PRINT"■"
150 LOCATE2,21:PRINT#0"-STACKY-■TIME   0":"SCR"
160 WHILE STRIG(0):WEND
170 IF INKEY$(0)=E$ THEN "MENU"
180 IF STRIG(0) GOSUB"MOVE" ELSE 230
190  IF P(0)>M ELSE 160
200  LINE(0,8)-(39,11)," ",BF:LOCATE0,8:PRINT" "
210  IF M MOD2=1 THEN LOCATE0,11:PRINT" "
220  LOCATE4,9:PRINT#0"CONGRATULATIONS!";:PAUSE30:GOTO30
230 S=STICK(0):X=X-(S=4)*(X>0)+(S=6)*(X<M)
240 LOCATEX*2+20,N-2:PRINT#0" ";C$;" ":PAUSE1
250 LOCATE30,21:PRINT#0USING"####";(TIME MOD10000):GOTO170
260 LABEL"MOVE":Y=0:J=M
270 A=B(X,Y):IF A<0 THEN Y=Y+1:IF Y>M THEN 330 ELSE 270
280 IF T(J)<0 THEN 320
290  IF T(J)<>A THEN J=J-1:GOTO 280
300  IF P(J)<P(J+1) THEN P(J)=P(J)+1 ELSE BEEP:GOTO330
310  "ERS":LOCATEP(J)*2+U,J*2+N:Q=A:"PRT":GOTO330
320 "ERS":T(J)=A:P(J)=1:LOCATEP(J)*2+U,J*2+N:Q=A:"PRT"
330 RETURN
340 '
350 LABEL"ERS":B(X,Y)=-1:LOCATEX*2+22,Y*2+N:Q=-1
360 LABEL"PRT":PRINT#0MID$(" ●○■▒◆目+▚",Q+2,1):RETURN
370 LABEL"CLS":CONSOLE0,20:CLS:CONSOLE:RETURN
380 LABEL"SCR":IF N>2 THEN LOCATE0,N-3:PRINT" "
390 FOR I=0 TO M:LOCATEU,I*2+N
400 PRINT#0"|";SPC(M+1);"||";SPC(M+1);"|";:NEXT
410 LOCATEU,M*2+N+2:PRINT#0STRING$(2,"└"+STRING$(M+1,"─")+"┘")
420 FOR J=0 TO M:FOR I=0 TO M:LOCATEI*2+22,J*2+N
430 Q=C(I,J):B(I,J)=Q:"PRT":NEXT
440 T(J)=-1:P(J)=0:NEXT:P(M+1)=M+1:X=0:TIME=0:RETURN
450 LABEL"MENU":"CLS":LOCATE8,4:PRINT#0"TRY AGAIN"
460 LOCATE8,8:PRINT#0"CORRECT ANSWER"
470 LOCATE8,12:PRINT#0"MODE CHANGE":Q=1
480 WHILE STRIG(0)=0:S=STICK(0):Q=Q-(S=8)*(Q>1)+(S=2)*(Q<3)
490 PRINT#0" ":LOCATE4,Q*4:PRINT#0"*":LOCATE4,Q*4:PAUSE1:WEND
500 IF Q=1 THEN 140
510 IF Q=2 THEN "ANS"
520 WHILE STRIG(0):WEND:REPEAT
530 S=STICK(0):IF S>2 AND S<9 THEN M=S-1
540 LOCATE18,12:PRINT#0USING"#×#   ",M+1,M+1:UNTIL STRIG(0)
550 BEEP:GOTO30
560 LABEL"ANS":"CLS":"SCR":FOR I=(M+1)^2-1 TO0 STEP-1:X=W(I)
570 LOCATEOX,N-2:OX=X*2+22:PRINT#0" ":LOCATEOX,N-2:PRINT#0C$
580 PAUSE5:"MOVE":NEXT:PAUSE30:GOTO450
```

(で)のぱーてぃハンズ──────(その4)

いよいよタマが撃てるぞ!

さあ，だんだんシューティングゲームっぽくなってきますよぉ。いよいよタマを撃ちます。にゃー。なんちて。ぽっくん（これこれ，人の原稿に落書きをするなって)。

ところで，ところで。タマを撃つにはどういうことが必要か？　ま，ジョイスティックのボタンを押すとタマが出る。ま，これはあたりまえなんですが，たとえばシューティングゲームでタマが出てるあいだ自機が動けなかったりしたらどうなるでしょう？　まず，ゲームにならないですよねえ。そう，ゲームっていうのは自分もタマも敵も一緒に動いてるんです。だから，今回はこういうプログラムの構造にします。

1) いま画面上にあるタマをさらに動かす
2) ジョイスティックのボタンが押されているかどうかを見る
3) ボタンが押されていてタマがまだ出せるようならタマを新しく出す

こんな感じのサブルーチン（関数と呼ぶのはなんか違和感があるなー）を作ります。

で，ミソは 1）の「いま画面上にあるタマを動かす」の部分なんです。普通はコンピュータは一度にひとつのことしかできないものなんですが，それではゲームにならん。そこで，一度にいろいろ動いてるように見せるために，

　自機，タマ，自機，タマ……，

と順繰りに動かしてやるんです。だから自機を動かすルーチンにこのサブルーチンを呼び出す行を1行つけ加えてやれば，自機とタマが同時に動いてるように見えるっつーわけです。ちなみに，敵と敵のタマが出てくると自機，タマ，敵，敵のタマ，自機，タマ……というふうに動かすわけです。わかりますよね。

そうそう，順繰りにっていえば，世の中にはワークステーションというえっらーいコンピュータがありまして，UNIXとかCとかが好きな人はおもわず「ははあっ」とひれ伏してしまうような（もちろん値段もむちゃくちゃ高い）ものだったりするんですね。で，これの特徴のひとつに「マルチタスク・マルチユーザー」という機能がありまして，ようするにこれって大勢の人がその1台のコンピュータでいっぺんにプログラムを動かすことができるんですよね。そう，ワークステーションっていうのはまるでコンピュータの聖徳太子のようなありがたーいものなんです。

……なんですが，そういうコンピュータでも中身は実はそんなにX68000とかと基本的にはたいした差はないものなんですよね（といっても，もちろん値段が高いだけあってスピードとかは桁違いに凄いけど)。実際，私の学校にもSUN3っていうありがたーいワークステーションがあるんですけど，実はCPUは68000の兄貴分のMC68020っていうやつだしメモリに至っては4Mバイトしかないから，人によっては自分のX68000のメモリのほうが多いって人もいるんじゃない？　っていうぐらいのものなんです。でもそんな機械でなんでそんなことができるかっていうと，実はワークステーションの基本ソフト（これがUNIXというやつなのだ）がいろんなプログラムを順繰りにやるようにコンピュータを操ってるだけだったりするんですよね，実は。ま，正確にはひとつの作業ごとにやっているわけではなくて，タイムシェアリング(TSS)といってプログラムごとに一定の時間を割り振って実行していたりするんですけど。

ほかにもたとえばX68000でもゲームのキャラクターを動かしながらFM音源でじゃかじゃか音を鳴らしてたりするでしょ。あれも本当はキャラクターを動かしてキャラクターを動かして……ほい，時間だFM音源を鳴らして「ジャン！」，キャラクターを動かして，ほい，時間だFM音源を鳴らして「ジャジャン！」……っていうぐあいにちょっとずついろんなことをするようにしてるんですよね。

しかし，なんだな。なんか今回は関係ない話ばっかしとるな。でも，実は初心者なのにFM音源の話をするときに「FM音源って割り込みかけて一定間隔で作業をしてるだけで別にいっぺんにグラフィックと音楽をやってるわけじゃないんだよね。タイムシェアリングシステムっぽいっていう感覚かな」なんてさらっと言っておお凄いととっても尊敬されてしまいました，なんていう感謝のハガキが来るかもしれないしな。ま，よしとしよう。さらに「やっぱりマルチタスクはTSSだよね。イベントドリブンなんておもちゃ，おもちゃ」と言ってしまえばさらにグッドだな。そして，何かわからないことを質問されたら「やあ，最近SUNしか使ってなくて」とか言って逃げられれば完璧！　かもしれない（おいおい，これじゃショートプロぱーてぃじゃなくて「初心者のためのパソコン見栄講座」じゃないか)。

ま，それはいいとして実際のプログラムを見ていきましょう。自機，タマを順繰りに動かすためには……。

まず，先月の160行から340行を見てください。自機の移動ルーチンになってますよね。そこの320行に，

　firemove( )

これです。あとはサブルーチンを作ればいいだけ，簡単でしょ。で，今回はゼビウスっぽくしようと思い，タマを1画面に3つ出せるようにしたかったんですよね。そこでタマの座標を，

　fire_x(0〜2)
　fire_y(0〜2)

という配列にしました。380行のfor文で3つのタマそれぞれの作業をするようにしています。……といってもスプライトを上に動かすだけだから，

　y=y−8

して，

　sp_set(……)

すりゃ終わりなんだよね。

410行と450行に注釈が入ってますが，ここでタマが敵に当たったかどうかの判定をしているんですね。

あと，注意するところは390行でジョイスティックを調べる (strig(0)) のがちょっと複雑になってるけど，これはジョイスティックが押しっぱなしになっているときにタマが連射にならないようにってんでつけたもんです。

とにかく，今回は座標を調べてループをかませてってだけですからねー，そんなに難しくなかったでしょ？　それ以前に内容がなかったという話もあるけど，そんなことは気にしない気にしない。ね，編集担当さん(いまごろ，これ読んで，怒りにぷるぷる震えてるだろーなー)。

ま，今月はこんなところかな。来月またこのOh!Xで。シュワッ！（と空を飛んで去る)

```
320 firemove()
370 func firemove()
380 for b=0 to 2
390   trig=strig(1) and trig:k=strig(1)
400 if fire_x(b)>0 then {
410  /*  atarif()   当たり判定が入る    */
420    if fire_y(b)<>0 then fire_y(b)=fire_y(b)-16
430    if fire_y(b)<16 then fire_x(b)=0  /*タマが上端（Y=0）の処理*/
440    sp_set(34+b,fire_x(b),fire_y(b),&H120)
450  /*  atarif()       */
460    } else {
470    if trig=0 and k=1 then{
480    fire_x(b)=x:fire_y(b)=y-8:trig=1:sp_set(34+b,fire_x(b),fire_y(b),&H120)
490 }
500 }
510   next
520 endfunc
```

アフターケア

創刊8周年記念PRO-68Kのその後

先月号の付録ディスク「創刊８周年記念PRO-68K」はいかがでしたか。内容の密度はどこにも負けないつもりですが，いろいろと説明の不備があり，編集部には問い合わせが殺到しています。一応これまでにわかったことをまとめてみました。

編集部

ディスク解凍(展開)方法について

●Human68kのVer.1をご利用の場合

各プログラムを自動的に解凍するバッチ処理（DISK.BAT）はHuman68kVer.1では動作しないことがわかりました。残念ながらまとめて解凍するにはかえって面倒な処理が必要となります。申し訳ありませんが，６月号49ページの表１を見て必要なものから個別に解凍してください。

●Human68kのVer.2をご利用の場合

DISK.BATを使ってまとめて解凍する場合，SMPL（サンプリング音を集めたもの）が解凍されずに残ってしまいます。これを解凍するには，３枚目ディスクへの展開が終わった状態（A：に付録ディスク，B：にユーザーディスク）で，次のように打ち込んでください。

```
A>B:
B>RD SND
B>MD SMPL
B>CD SMPL
B>A:LH -E A:SMPL
```

これにより，３枚目のディスクにSMPLというディレクトリを作り，その中にサンプリングデータを展開することができるはずです。先月号49ページの解説には下線部のA：が抜けていました。申し訳ありません。

◆

Q：付録のディスクを解凍しようとしたら「主記憶が不足しています」というエラーメッセージが出てしまいます。メモリを増設しなければ解凍できないのでしょうか。

A：ディスクを解凍するだけなら，増設の必要はありません。ただし，この春発売されたX68000PROIIシリーズの場合，システムディスクをそのまま使ってビジュアルシェルからコマンドモードを起動すると，展開のためのメモリが足りなくなってしまうことがわかりました。

メモリを確保するには，システムディスクをコマンドモードから立ち上げるように書き換えるとよいでしょう。システムディスクのCONFIGというディレクトリの中にあるTITLE.SYSをTITLE.VSのように変更（Rename）してください。次回からはコマンドモードから立ち上がるはずです。詳しくは取扱説明書の第３部「3.4起動時にコマンドモードに入るには」(165ページ)をご覧ください。

各プログラムの利用について

付録ディスクにはシステムは入っていません。収録されているプログラムは単独では使えず，それぞれ必要に応じたシステムで起動しなければなりません。

また，X68000は機種によって付属のシステムディスクの内容が微妙に違いますし，ユーザーの使用状況によって対応のしかたも違ってきます。初心者の方にはわかりにくいことと思いますが，以下にだいたいの目安を示しますので参考にしてください。

●デバイスドライバの登録

基本的に，標準のシステムディスクは通常必要なほとんどのドライバが登録されていますが，機種によってその内容は異なる場合があります。たとえば，FM音源ドライバとして現在標準となっているOPMDRV.Xと浮動小数点演算パッケージFLOAT2.Xは，初代機種のシステムディスクには入っていませんでした。

OPMDRV.XとFLOAT2.Xはその後発売された数多くの市販ソフトに入っていますのでたいていの人はお持ちだと思いますが，これがないと，付録ディスクのYet Another Columnなども実行することができません。お持ちでない方はこの機会にHuman68kのVer.2.0を買うことをおすすめします。もっと安くすむ方法としては満開製作所発行のディスクマガジン電脳倶楽部を購読するという手もあります。事情が明記されたものに限り特別に１カ月（1,000円）だけでも応じてくれるそうです。

デバイスドライバを組み込むには，システムのなかにある，config.sysを書き換えます。一般にはED.Xというエディタを利用しますが，ここでは日本語ワードプロセッサWP.Xを使ってみましょう。

ワープロが起動したら，書き換えたいシステムディスクをAドライブに入れ，マウスで「ファイル入力」を選びます。ここでconfig.sysを読み込み，

DEVICE＝A:¥OPMDRV.X

のように登録したいドライバのファイル名とその位置（この場合Aドライブのルートディレクトリ）を指定する１行を書き加えます。あとは「ファイル出力」を選んでconfig.sysを選択し，OKで上書きすれば終了。リセットで書き換えられたシステムが起動します。

逆に，ドライバを外す場合には，

＊DEVICE＝A:¥OPMDRV.X

のように＊をつけるとそのドライバは無効になります。

●メインメモリの問題

メインメモリの空き容量は利用状況によって異なります。

たとえば，PROII/EXPERTIIに付属のシステムディスクは同じHuman68kのVer.2でも，BIOSを高速化するIOCS.Xが登録されているためシステムの使用するメモリがそのぶん多くなっています。

このように，登録されているドライバによっても必要となるメモリが変わってきます。メモリに余裕のない場合はできるだけ不要なドライバを外してご使用ください。

●専用ディスクの作り方

各プログラムをシステムディスクに組み込んで使うにはディスクに十分な空き容量が必要です。お持ちの機種によってシステムディスクの空き容量はかなり違いますが，とりあえず，BINというディレクトリを削除すれば，500Kバイト以上の容量が確保できるはずです。

ウイルス検出プログラム

さて，X68000のウイルスですが，あなたのマシンとディスクは大丈夫だったでしょうか。問題のウイルスは，７月になると発病し大切なディスクのファイルを破壊する恐れがあります。まだ，チェックを行っていない方はぜひとも早めにチェックを行ってください。また，いまのところウイルスには冒されていないという方も予防のため，しばらくはDOCTOR.Rを組み込んでおくことをおすすめします。

Q：Oh!Xのワクチンは SRAMを使うから危険だという話を聞きましたが本当に大丈夫なのでしょうか？

A：SRAM(スタティックRAM)を使うから危険ということはありません。また，X68000はSRAMがあるからウイルスに感染しやすいなどという人もいるようですがこれはナンセンスです。SRAMは比較的コストが高いので16Kバイトと小さな容量しかありませんが，ユーザーにも開放された便利な機能のひとつです。今回のウイルスはSRAMに常駐するタイプなので，ユーザーの皆さんが独自の用途に使っている場合，ウイルスの侵入でSRAMが書き換えられても即座に気づくことでしょう。

一般に，ワクチンは個別のウイルスに対してのみ有効で，どんなウイルスにも効く万能ワクチンというものはありえません。しかし，SRAMにDOCTOR.Rを組み込んでおけば，少なくともIPL型のウイルスの侵入は防ぎやすくなりますし，SRAM常駐型のウイルスなら，万一侵入を許してもワクチン自体が破壊されるため発見が容易になります。

Oh!Xでは昨年6月号で発表したウイルスチェッカをSRAMに組み込んで使っていました。そもそも，今回のウイルスが確認されたのはこのウイルスチェッカが破壊されたためです。

今回のウイルスに手を加えた亜流が出てくる可能性も考えられますので，現在SRAMを使用していない方は当面DOCTOR.Rを組み込んだ状態で使用してくださるようご協力をお願いいたします。

Q：DOCTOR.Rを組み込んだのですが，ドライブになにもディスクを入れないで起動しようとすると，「IPLをチェックしています。セルフチェックOK」「ドライブ0にHumanのディスクを入れてください」のあと「エラーが発生しました。リセットしてください」と出るのです。なぜでしょう。私のX68000がおかしいのでしょうか？

A：心配ありません。というよりエラーとなるのは正常な証拠です。DOCTOR.Rを組み込んでおくと，電源ON(またはリセット) でシステムが起動する前にそのディスクがウイルスに冒されていないかチェックするようになっています。しかし，ディスクが準備できていない場合，ディスク挿入と同時に起動させてしまうとウイルスの侵入を許すことになってしまいます。そこで，もう一度リセットをかけてディスクをチェックするようわざとエラーを発生させているのです。

ただし，ふつうのハードディスクをつないでいる場合，ディスクが準備されていなければ素直にハードディスクから起動するため，エラーとはなりませんし，感染の恐れもありません。また，SCSIタイプのハードディスクを使っている場合は起動をSCSIデバイス1にしないと立ち上がらないようなので，DOCTORはコマンドラインからの使用のみでウイルスチェックをしなければなりません（TFR2は使わないでください)。

また，SRAMを初期状態に戻すには，

A>SWITCH B=ROM0
A>SWITCH B=STD

と打ち込んでください。

Yet Another Column

このゲームはBASICで作成されたものをコンパイルしたものです。プレイするには，FM音源ドライバOPMDRV.Xと浮動小数点パッケージ“float2.x”（または“float2+”)を組み込んだシステムディスクを使用してください。また，実行ファイルは“yet.x”，背景のグラフィックデータは“siro.gs3”です。

とりあえずゲームを楽しみたい方は，BINを削除したシステムディスクをAドライブで立ち上げ，Bドライブに付録ディスクを入れます。コマンドモードで，

A>B：LH－E B：GAMES

としてください。展開が終了したら，

A>YET

と打ち込めばゲームが始まります。以後はビジュアルシェルでyet.xのアイコンをダブルクリックしてもOKです。

音楽演奏関係

Q：ナイトアームズのエンディングテーマの演奏方法がよくわかりません。

A：53ページ右下のOPMDサンプル曲の説明にはいろいろと不備がありました。

単独でOPMDとSMPLを解凍する場合，まず，BASICの入った500Kバイト以上空きのあるシステムディスクをAドライブ，オマケディスクをBドライブに入れ，コマンドモードから，

A>B:LH －E B:OPMD
A>B:LH －E B:SMPL

としてください。これは先月の囲み記事のとおりです。

また，DISK.BATでまとめて解凍した方は，SMPLを単独で解凍したあと，用意したシステムディスクにOPMDとSMPLの中身をすべてコピーしてください。システムを作成するディスクをAドライブ，3枚目の解凍ディスクをBドライブに入れ，

A>COPY B：OPMD
A>COPY B：SMPL

と入力します。

内蔵音源のみの場合なら，

A>OPMD /A BOS

MIDIでM1を鳴らすなら，

A>OPMD /M KAEND

同じくMT-32の場合，

A>OPMD /M KEMT

として，OPMD.Xを立ち上げます。

あとは，

A>BASIC

でBASICを立ち上げ，

LOAD“KAEND.BAS”
RUN

でOKです。

Q：MUSICDRVで設定した音色と変更された音色が違うのですが……。

A：MUSIC2.FNCを使うと音色登録時に番号がひとつずれてしまいます。以下の修正を加えてください。まず，

A>MAC

としてTOOLSに入っていたマシン語入力ツールを起動します。NEW FILEかどうかにはNと答えて，ファイル名にMUSIC2.FNCを指定してください。

プログラムが読み込まれたら，“E”でエディットモードに入り，

0451　53　→　4E
0452　80　→　71
0457　80　→　81
0495　53　→　4E
0496　80　→　71
049C　80　→　81

のようにデータを書き換えてください。この6カ所を変えればあとはSキーを押してセーブし「！」で処理を中止します。

また，MUSIC.DRVはOPMDRVと同時には使えません。

そのほかのプログラムについて

Q：PIC.FNCを組み込むとBASIC起動時にエラーが出るのですが。

A：ファイル内のデータが2バイトずれていました。原因はよくわかりません。とりあえず，PIC.FNCをPICO.FNCにリネームしたうえでBASICからリスト1を入力，実行してください。また，コンパイルして使う場合は，各関数のパラメータは省略しないようにしてください。

リスト1

```
 10 /* ***************PIC.FNC debug
 20 /* frename("pic.fnc","pic0.fnc")
 30 char a(5000)
 40 int i,j,k
 50 /*
 60 i=fopen("pic0.fnc","r")
 70 j=fopen("pic.fnc","c")
 80 fread(a,&H400,i)
 90 fwrite(a,&H400,j)
100 fputc(&H2F,j):fputc(&H3C,j)
110 fread(a,4724-&H400,i)
120 /*
130 for k=0 to 36
140   a(k*4+&H16F)=a(k*4+&H16F)+2
150 next
160 a(&HE17)=a(&HE17)+2
170 fwrite(a,4724-&H400,j)
180 fcloseall()
```

Q：もしかしてPurePASCALにはアセンブラが必要なのでしょうか。

A：特に触れていませんでしたが，PurePASCALのコンパイラはアセンブラのソースを出力する形式をとっています。したがって，実行ファイルの作成にはアセンブラおよびリンカ（AS.X，LK.Xなど）が必要です。PurePASCALのシステムディスクを作成する場合はこれらを忘れずに組み込んでください，なお，コンパイルスイッチをなにも指定しなければ，*.s，*.o，*.xの順で実行ファイルを自動的に作成するようになっています。詳しくはドキュメントPASCAL.MANをご覧ください。

Q：PurePASCALでFLOAT2+/3+が正常に動きません。

A：現在調査中です。FLOAT2+/3+は使わないでください。

Q：ディスクに入っていた99.BASはどうやって起動すればいいのですか。

A：99.BASはBASIC用CARD.FNCのサンプルです。CARD.FNCの組み込み方がわからない人は以下にその1例を書きましたのでこのとおりに実行してみてください。

1) まず，システムディスクをコピーします（AドライブからBドライブへ)。

注）システムディスクの空き容量が135Kバイト以上ない場合はコピーしたあとコマンドモード上で，

```
DEL B:BIN
COPY A:¥BIN¥ED.X B:
```

としてBINを殺してディスク容量を確保します。

2) コピーしてできたディスクをAドライブに，オマケを展開したディスク（GAMESというディレクトリが入っているもの）をBドライブに入れます。

3) BASICと打ち込んでBASICを起動します。

4) 以下のコマンドを実行してください。

```
CHDRV"B:"
CHDIR"GAMES"
LOAD"MAKE.BAS"
RUN
```

これでデータが展開されて，CARD.FNCというファイルが出来上がります。

5) 次に，SYSTEMと打ち込みコマンドモードに戻ります。

6) ここで，

```
COPY B:¥GAMES¥CARD.FNC A:BASIC2
```

（初代X68000についてきたシステムディスクの場合はBASIC2のところをBASICにする）として，BASICのディレクトリにCARD.FNCをコピーする。

7) コマンドモードで

```
A>ED ¥BASIC2¥BASIC.CNF
```

として，エディタを起動します（さっきと同じように，BASIC2じゃない場合はBASICに変える)。

8) 画面にBASIC.CNFの内容が表示されますので，そこに，

```
FUNC=CARD
```

と1行書き加えてください。

9) ESC+Eを押してセーブします。

以上でCARD.FNCのBASICへの組み込みは完了です。

Q：DIS.Xの解説にあったFEFUNC.H はディスクには入っていませんがどうすればいいのでしょう。

A：DISが起動時に参照するインクルードファイルのうちFEFUNC.Hはディスクにはついていません。C compiler PRO-68Kにはついてくるのですが，アセンブラ（THE福袋など）しかお持ちでない方はエディタからリスト2を入力してください。

また，FEFUNC.Hという名前のファイルさえ用意しておけばDIS.Xを使用することはできます。

なお，DOSCALL.MAC，IOCSCALL.MAC，FEFUNC.Hはいずれも C compiler PRO-68Kから転載させていただいたものです。

リスト2

```
==================  FEFUNC.H  ==================
   1:   .nlist
   2: *
   3: * fefunc.h X68k XC Compiler v1.01 Copyright 1987 SHARP/Hudson
   4: *
   5: FPACK     macro   callname
   6:   dc.w    callname
   7:   endm
   8:
   9: ****************************************
  10: __LMUL          EQU     $FE00           d0=d0*d1
  11: __LDIV          EQU     $FE01           d0=d0/d1
  12: __LMOD          EQU     $FE02           d0=d0 mod d1
  13: *               EQU     $FE03
  14: __UMUL          EQU     $FE04           d0=d0*d1
  15: __UDIV          EQU     $FE05           d0=d0/d1
  16: __UMOD          EQU     $FE06           d0=d0 mod d1
  17: *               EQU     $FE07
  18: __IMUL          EQU     $FE08           d0d1=d0*d1
  19: __IDIV          EQU     $FE09           d0ｱﾏﾘd1=d0/d1
  20: *               EQU     $FE0A
  21: *               EQU     $FE0B
  22: __RANDOMIZE     EQU     $FE0C           err=d0(0~65535)
  23: __SRAND         EQU     $FE0D           err=d0(-32768~32767)
  24: __RAND          EQU     $FE0E           d0=rand()
  25: *               EQU     $FE0F
  26: **************************************
  27: __STOL          EQU     $FE10           d0=(a0).１０進
  28: __LTOS          EQU     $FE11           (a0)..=d0
  29: __STOH          EQU     $FE12           d0=(a0).１６進
  30: __HTOS          EQU     $FE13           (a0)..=d0
  31: __STOO          EQU     $FE14           d0=(a0).８進
  32: __OTOS          EQU     $FE15           (a0)..=d0
  33: __STOB          EQU     $FE16           d0=(a0).２進
  34: __BTOS          EQU     $FE17           (a0)..=d0
  35: __IUSING        EQU     $FE18           (a0)..=d0,d1桁
  36: *               EQU     $FE19
  37: *************************************
  38: __LTOD          EQU     $FE1A           d0d1=d0
  39: __DTOL          EQU     $FE1B           d0=d0d1
  40: __LTOF          EQU     $FE1C           d0=d0
  41: __FTOL          EQU     $FE1D           d0=d0
  42: __FTOD          EQU     $FE1E           d0d1=d0
  43: __DTOF          EQU     $FE1F           d0=d0d1
  44: **************************************
  45: __VAL           EQU     $FE20           d0d1=(a0).16/10/8/2進
  46: __USING         EQU     $FE21           (a0)..=d0d1,d2.d3,d4
  47: __STOD          EQU     $FE22           d0d1,d2,d3=(a0)１０進
  48: __DTOS          EQU     $FE23           (a0)..=d0d1
  49: __ECVT          EQU     $FE24           (a0),d0,d1=d0d1,d2全体桁
  50: __FCVT          EQU     $FE25           (a0),d0,d1=d0d1,d2小数点桁
  51: __GCVT          EQU     $FE26           (a0)..=d0d1,d2全体桁
  52: *               EQU     $FE27
  53: **************************************
  54: __DTST          EQU     $FE28           z=d0d1
  55: __DCMP          EQU     $FE29           z,c=d0d1-d2d3
  56: __DNEG          EQU     $FE2A           d0d1=neg(d0d1)
  57: __DADD          EQU     $FE2B           d0d1=d0d1+d2d3
  58: __DSUB          EQU     $FE2C           d0d1=d0d1-d2d3
  59: __DMUL          EQU     $FE2D           d0d1=d0d1*d2d3
  60: __DDIV          EQU     $FE2E           d0d1=d0d1/d2d3
  61: __DMOD          EQU     $FE2F           d0d1=d0d1 mod d2d3
  62: ***************************************
  63: __DABS          EQU     $FE30           d0d1=abs(d0d1)
  64: __DCEIL         EQU     $FE31           d0d1=ceil(d0d1)
  65: __DFIX          EQU     $FE32           d0d1=fix(d0d1)
  66: __DFLOOR        EQU     $FE33           d0d1=floor(d0d1)
  67: __DFRAC         EQU     $FE34           d0d1=frac(d0d1)
  68: __DSGN          EQU     $FE35           d0d1=sgn(d0d1)
  69: __SIN           EQU     $FE36           d0d1=sin(d0d1)
  70: __COS           EQU     $FE37           d0d1=cos(d0d1)
  71: __TAN           EQU     $FE38           d0d1=tan(d0d1)
  72: __ATAN          EQU     $FE39           d0d1=atan(d0d1)
  73: __LOG           EQU     $FE3A           d0d1=ln(d0d1)
  74: __EXP           EQU     $FE3B           d0d1=exp(d0d1)
  75: __SQR           EQU     $FE3C           d0d1=sqr(d0d1)
  76: __PI            EQU     $FE3D           d0d1=pi()
  77: __NPI           EQU     $FE3E           d0d1=pi(d0d1)
  78: __POWER         EQU     $FE3F           d0d1=pow(d0d1,d2d3)
  79: __RND           EQU     $FE40           d0d1=rnd()
```

```
 80: *                      EQU     $FE41
 81: *                      EQU     $FE42
 82: *                      EQU     $FE43
 83: *                      EQU     $FE44
 84: *                      EQU     $FE45
 85: *                      EQU     $FE46
 86: *                      EQU     $FE47
 87: *                      EQU     $FE48
 88: __DFREXP               EQU     $FE49         D0D1,D2=D0D1
 89: __DLDEXP               EQU     $FE4A         D0D1=D0D1,D2
 90: __DADDONE              EQU     $FE4B         d0d1=d0d1+1#
 91: __DSUBONE              EQU     $FE4C         d0d1=d0d1-1#
 92: __DDIVTWO              EQU     $FE4D         d0d1=d0d1/2#
 93: __DIEECNV              EQU     $FE4E         d0d1=d0d1
 94: __IEEDCNV              EQU     $FE4F         d0d1=d0d1
 95: ****************************************
 96: __FVAL                 EQU     $FE50
 97: __FUSING               EQU     $FE51
 98: __STOF                 EQU     $FE52
 99: __FTOS                 EQU     $FE53
100: __FECVT                EQU     $FE54
101: __FFCVT                EQU     $FE55
102: __FGCVT                EQU     $FE56
103: *                      EQU     $FE57
104: ****************************************
105: __FTST                 EQU     $FE58
106: __FCMP                 EQU     $FE59
107: __FNEG                 EQU     $FE5A
108: __FADD                 EQU     $FE5B
109: __FSUB                 EQU     $FE5C
110: __FMUL                 EQU     $FE5D
111: __FDIV                 EQU     $FE5E
112: __FMOD                 EQU     $FE5F
113: ****************************************
114: __FABS                 EQU     $FE60
115: __FCEIL                EQU     $FE61
116: __FFIX                 EQU     $FE62
117: __FFLOOR               EQU     $FE63
118: __FFRAC                EQU     $FE64
119: __FSGN                 EQU     $FE65
120: __FSIN                 EQU     $FE66
121: __FCOS                 EQU     $FE67
122: __FTAN                 EQU     $FE68
123: __FATAN                EQU     $FE69
124: __FLOG                 EQU     $FE6A
125: __FEXP                 EQU     $FE6B
126: __FSQR                 EQU     $FE6C
127: __FPI                  EQU     $FE6D
128: __FNPI                 EQU     $FE6E
129: __FPOWER               EQU     $FE6F
130: __FRND                 EQU     $FE70
131: *                      EQU     $FE71
132: *                      EQU     $FE72
133: *                      EQU     $FE73
134: *                      EQU     $FE74
135: *                      EQU     $FE75
136: *                      EQU     $FE76
137: *                      EQU     $FE77
138: *                      EQU     $FE78
139: __FFREXP               EQU     $FE79
140: __FLDEXP               EQU     $FE7A
141: __FADDONE              EQU     $FE7B
142: __FSUBONE              EQU     $FE7C
143: __FDIVTWO              EQU     $FE7D
144: __FIEECNV              EQU     $FE7E
145: __IEEFCNV              EQU     $FE7F
146:
147: __CLMUL                EQU     $FEE0
148: __CLDIV                EQU     $FEE1
149: __CLMOD                EQU     $FEE2
150: __CUMUL                EQU     $FEE3
151: __CUDIV                EQU     $FEE4
152: __CUMOD                EQU     $FEE5
153: __CLTOD                EQU     $FEE6
154: __CDTOL                EQU     $FEE7
155: __CLTOF                EQU     $FEE8
156: __CFTOL                EQU     $FEE9
157: __CFTOD                EQU     $FEEA
158: __CDTOF                EQU     $FEEB
159:
160: __CDCMP                EQU     $FEEC
161: __CDADD                EQU     $FEED
162: __CDSUB                EQU     $FEEE
163: __CDMUL                EQU     $FEEF
164: __CDDIV                EQU     $FEF0
165: __CDMOD                EQU     $FEF1
166: __CFCMP                EQU     $FEF2
167: __CFADD                EQU     $FEF3
168: __CFSUB                EQU     $FEF4
169: __CFMUL                EQU     $FEF5
170: __CFDIV                EQU     $FEF6
171: __CFMOD                EQU     $FEF7
172: __CDTST                EQU     $FEF8
173: __CFTST                EQU     $FEF9
174: __CDINC                EQU     $FEFA
175: __CFINC                EQU     $FEFB
176: __CDDEC                EQU     $FEFC
177: __CFDEC                EQU     $FEFD
178:
179: __FEVARG               EQU     $FEFE   >d0,d1
180: __FEVECS               EQU     $FEFF   d0,a0>d0
181:         .list
```

突然ですが
創刊100号記念の予告です
激突 祝一平VS西川善司

若手実力ナンバーワンの西川善司が満開製作所の祝一平氏に挑戦状をFAX送信！ ☞

通信ケーブルに炸裂する地震，洪水，火山
そして毒の沼に落ちるのはいずれか？
編集室は闘いのワンダーランドと化す

［注意事項］

Oh！X1990年8月号は諸々の事情により定価据え置きの560円(税込)となります。要するにオマケはつきません。あらかじめご了承ください。

通巻100号のなかには，本誌別冊の「ADVANCED MZ-700」(通巻35号)も含まれております。あらかじめご了承ください。

なお，次号では豪華なプレゼント企画が予定されています。あらかじめご了承ください。

たいへん申し訳ありませんが，一部お見苦しいページのある場合が考えられます。あらかじめご了承ください。

第49回

猫とコンピュータ

ホットラインで

Takazawa Kyoko
高沢 恭子

盆，正月に次いで一大イベントと化したゴールデンウィーク。今年のキョウコさん一家は，パソ通仲間と電話で親睦を深めたり，近場に出かけたりなど，ゆったりと休日を満喫したようですね。

「メールなんですけどね。送ってしまってから内容の修正をしたいとき，どうすればいいんでしょう」

「メールですか？　メールは……」

「あの，PC-VANなんか，まちがえたときはサブメニューが出て訂正できるようになってるんですよね」

「あ……，FBIはそういうコマンドはありませんねえ……あらためて送りなおすしかないんです。私もそうしてますけど」

「そうですかぁ，あと，アップロードした自分の書き込みを消したいときなんですけど。このあいだご主人にうかがったんですが忘れてしまいましてね」

「あれはタイトルの頭にアスタリスクををつけて，CRすればいいそうですよ」

「いやぁ，やってみたんですが，まったく受けつけないんです」

「じつは私も前に失敗したもので，使ってないんです。主人が帰りましたらすぐお電話さしあげます」

「いや，どうもどうも」

熟年のパソコンマニア，コバヤシカズヨシさんが半年ほど前から通信をはじめられて，今日もあいにく夫の帰宅前に質問の電話をいただいた。パソコンのご趣味は長く，ずっとひとりでいじっておられたが，1年くらい前に日経パソコンに掲載されたクラブ紹介の記事を見て「きまぐれコンピュータクラブ」に入会された。ここでクラブのボードがあるFBIにも加入されたのだ。

パソコンにくわしい人でも，通信はネットごとにマニュアルがさまざまなので，わかりにくいこともたくさんある。そんなときはコバヤシさんのように，さっさとメンバーに聞いてみるのがいちばんいい。でも，パソコン族はこういうダイレクトな手段をえらばない人が多いみたいだ。

帰宅した夫にコバヤシさんの質問を伝えると，彼はすぐにFBI-NETにアクセス。「きまこん」のボードにひとつ書き込みをしたあと，「消去」の実演をしてダウンロード。その実録を入れた手紙をコバヤシさん宛にMAILした。それから電話をして，MAILBOXを見てくださいと伝えた。

ホストが消えた

「あの……またちょうど仙台に出かけてまして」

なぜかコバヤシさんからのお電話に，夫はいつも不在だ。

「そうですか。いや，FBIがこのあいだの日曜日から通じないんです」

「あ，いまシスオペのお宅はお引っ越しの直後なので，システムは閉じてるんです。CLOSEは1週間の予定だそうですけど」

「そうなんですかぁ。コバヤシ先生もつながらないってふしぎがってましてね」

同じコバヤシさんでも，「目白のヤス」ことコバヤシヤスジ先生は大学の先生で，こちらのカズヨシさんは元お役人だ。

「広報室のボードに『お知らせ』があったんですけど……」

「だって，そんなとこ見ませんもの」

ちょっと不満そうにされたのが，とても率直でかえって好感をもってしまう。

「新しいインテリジェントビルの誕生です。ナカムラシスオペ個人のお宅ですけどきれいな白いビルだそうです」

「そりゃぁ，花束でも持っていかなきゃいけませんな，ハハハ」

明るい声に，私も電話のこちら側でうれしくなってきた。なんだかFBI-NETそのものがバージョンアップして新装開店するような気分だ。

当のナカムラ隊長のお宅では，そう浮かれてばかりはいられない大移転のようだった。ゴールデンウィークの過ごし方は，ニュース番組の主要なタイトルのひとつにもなっていたが，隊長一家の黄金週間はなんと引っ越しだったわけだ。5回線ぶんの設備の移動だけでもたいへんなのはたやすく想像できるが，その上，生活一式を運ぶのだ。じっさい，段ボール30箱ほどつめても，まだマシンが入りきらないとは，準備中の隊長の弁だった。

23番めの区

わが家のゴールデンウィークは，「わが家を基地にしてゆったりと過ごす」のがテーマだった。当初の予定だった北海道旅行は早いうちにキャンセルした。これは実行していれば雨の日に当たっていたので，運のよい判断だったような気もするけれど，出かけた場合のお天気はまた違うのかもしれない。

トオルの希望で銀座のドイツ料理を味わったり，近くの葛西臨海公園に遊んだりした。公園内にある話題の東洋一という水族館も，お友だちと出かけたことのあるトオルの推薦だったが，休日の超人気で入館はあきらめなければならなかった。

千葉県にもっとも近い東京の東端のわが区は，新しい都市としての発展がめざましい。都心が人の住まうところとしては廃墟になっていくかわりに，このあたりの人口はどんどんふくらんでいる。それにあわせて，区の町づくりの意欲はたいへんなものだ。

小学生のころ23区を暗記するとき，番号がついていた。1 千代田，2 中央，3 港，4 新宿，…この区は23番めだった。でも世田谷とならんで面積が大きい。だからいまこうして，公園，施設，広場，鉄道もでき

る。店舗の種類も数もじゅうぶんなうえに，さらに建設中のビルが，ひとつの視野に10あまりも見える。

わが家のあるN町のとなり，S町とR町の人口増加はたいへんらしい。川沿いから海に向けて広がっていこうとしている高層団地の町である。古い町を活性化させるのではなく，新しい町を創造していく人たちは，いまとても努力をしている。教育，文化，地域のコミュニティなど，町全体で高めておたがいをはぐくんでいこうとしている。ちなみにS町の団地内にある中学校のひとつは，都内一の学力を持っているのだそうだ。

ゴールデンウィークが明けてすぐ，この２つの町が主催した「ふれあいまつり」があった。団地内の子供会や，300以上もあるさまざまな趣味の同好会が力をあわせて，広場のプロムナードにたくさんの催しをくりひろげていた。リサイクルの店，伝承あそびのコーナー，健康相談。古本市，金魚すくい，伝統工芸の展示。手作りみこしや鼓笛隊のパレードもあり，石炭をたいたミニSLも走っている。その中で，NTTコーナーの立て看板が，「ひつじ年から都内の局番が４ケタになります」と，にぎにぎしさを象徴するように呼びかけていた。

運営委員会と思われる方たちはそろいのTシャツで，きびきびとスタッフをつとめている。どの人も，みずからが楽しんでいるようすで，これがおまつりを盛り上げるいちばんの力にちがいないと思った。きっとよく話し合いが持たれて，みんなが目標を共有しているからだろう。

空を見上げると，垂直にいくつもいくつも，マンションの棟がそそり立っている。こういったマンモス団地の場合は，となりの棟まで行くにもけっこう距離があるものだ。たぶんパソコンの同好会もあるにちがいないから，そのうち自治会活動もパソコン通信などを活用したものになっていくのだろう。

そのS町とR町は，いま人口急増のために，幼稚園，小学校がパンク状態になっているそうだ。以前に新設されたものもすでに満杯で，２つの町の父母たちは増設の運動を熱心にすすめているが，子供の数は確実に減っていくことを主張する区側はこばんでいる。１クラスの人数が多すぎて，お友だちの名前がおぼえきれないほどの幼稚園もあって，数が多すぎることの弊害が深刻にうったえられている。

ウイルスの害

ゴールデンウィークはX68000のウイルス報道も話題になった。用語だけは取り揃えてあったけれど，なんだか筋書きが未熟すぎるので，奇妙に思った人はたくさんいたと思う。

でも，パソコンに馴染みのない人にとっては，さぞものものしいできごとに映ったことだろう。自分の知らない世界については，どんな奇抜な想像があってもおかしくない。プログラムの「虫」というのは，バクテリアのようにほんとうにマシンを食いあらすと思っていた，とジョークまじりの新聞の投稿を読んでいても，とても共感できる。

私は逆に，むかし祖母や母が話す「衣類の虫」というのは，布が古びたりいたんだりすることだと思っていた。それがあるとき，ほんとうに生きた虫が衣類を食べて穴をあけてしまうのだと知って，びっくり仰天した。「本の虫」というのも，本好きの人のたとえに使われることもあって，じっさいに生きた虫も住みついていると知ったときは，まったく驚いた。

泣き虫，弱虫，カンの虫。「虫のいどころ」「虫が好かない」「虫の知らせ」なんていうものだから，衣類の虫も本の虫も抽象的なものだと思いこんでいたのだ。

コンピュータウイルスも，マシンにさわっていると感染すると信じている人がいるそうだけれど，解釈によっては，まったくのまちがいとはいえないと思う。

人間があやつることでパソコンは動き，ウイルスの汚染も人間が操作することで行われる。私たちは感染の経路にあるし，マシンとの一体感も持っている。パソコン通信などの，信頼をベースにしているはずのコミュニケーションからウイルスをうつされるとき，精神の痛手もバカにできない。これは人間も感染したと考えられないことはない。「ウイルス」(virus) を辞書でひくと，①「濾過性病原体」，②(道徳的腐敗の)もと，害毒，となっている。

シャイが主流？

「あのですね，13日OPENというのでアクセスしてみたんですけど，FBIはまだ通じないんですよ」

コバヤシさんからのお電話に，やっと夫が出ることができた。

「ああ，少し遅れているらしいです。その間の情報は，ナツメの『千夜一夜』のボードにだいぶ書かれてますから……。ええとアクセスポイントは……」

それから3日ほどして，例の明るい声。

「やっと復活しましたねぇ，前のとおりなのでホッとしました。ナツメは入ってみましたが，なんだかわからないことばかり書いてあって，ハハハ」

5月16日にFBIは再開した。ボートピープルの気分だったというメンバーを，一時的に収容したNATUME-NETの「千一」ボードは，一気に100以上のアーティクルが増えた。

10日ばかりのシステム停止で，私たちもホスト局のありがたさがよくわかったが，その一方，コバヤシさんからの何回かのお電話で，FBI不在のあいだもほのぼのと楽しかった。閉鎖的や自分本位の代名詞にされがちのパソコンマニアだけれど，ほんとはみんなコバヤシさんのようにやってみたいのだ。でも，ちょっとガマンしている。

BACK ISSUES バックナンバー案内

ここには1989年7月号から1990年6月号までをご紹介しました。現在1989年6～12，1990年1～6月号までの在庫がございます。バックナンバーおよび定期講読のお申し込み方法については，168ページを参照してください。

1989

7月号
特集 3Dグラフィックへの飛翔
Zバッファアルゴリズム/スムースシェイディング 他
THE SOFTOUCH Terazzo PRO-68K/アドヴァンスト・ファンタジアン
新連載 DōGA・CGアニメーション講座
MZ-2500用グラフィックエディタ作成講座
マシン語カクテル in Z80's Bar
X-BASICプログラミング調理実習
全機種共通システム TTC用パズルゲームTIC BAN
X68000マシン語プログラミング/C調言語講座PRO-68K 他

8月号
特集1 X1プログラミングガイドブック
PCGの基礎から奥義まで/超高速ラインルーチン 他
特集2 3Dグラフィックの深淵へ
スキャンラインZバッファ/3Dモデリング 他
新連載 (で)のショートプロぱーてぃ
X68000マシン語プログラミング/C調言語講座 PRO-68K
X-BASICプログラミング調理実習/DōGA・CGA講座
MZ-2500用グラフィックエディタ/Z80's Bar 他
全機種共通システム CP/M用ファイルコンバータ

9月号
特集 活用ハードディスク&プリンタ
各社ハードディスク接続総チェック/ハードディスク雑学講座/COPYキーメニュー/ビデオプリンタ活用プログラム 他
THE SOFTOUCH ジェノサイド/琉球/mFORTH Compiler
● サイバースティックで遊ぶ 不思議な環境ソフトの世界
● X1/X1turbo用シューティングゲーム Defeat X
Z80's Bar/MZ-2500グラフィックエディタ 他
[X68000] X-BASIC/マシン語/C調言語講座/DōGA・CGA
全機種共通システム 生物進化シミュレーションBUGS

10月号
特集 ゲーム面白心理学
ソーサリアン・宇宙からの訪問者/ファンタジーゾーン
ねじ式/ガウディ・バルセロナの風/サバッシュ 他
● MZ-700用シューティングゲームSide Roll-F
● X1/X1turdo用カードゲームBonding
ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/DōGA・CGA
THE SOFTOUCH Z'sTRIPHONY DIGITAL CRAFT/James68K
全機種共通システム 小型インタプリタ言語TTI

11月号
特集 microComputer入門
初歩からのCPU物語/RISCプロセッサの設計と製作
X68000&X1で周辺LSIを使いこなそう
連載 ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/DōGA・CGA
● X68000用カードゲームばばぬき
LIVE in '89 メタルホーク/オブ・ラ・ディ，オブ・ラ・ダ
THE SOFTOUCH Stationery PRO-68K/リングマスター1
全機種共通システム TTI用パズルゲームPUSH BON!

12月号
特集 Cプログラミングへの招待
付録 C言語簡易リファレンス
連載 ショートプロぱーてぃ/Z80's Bar
X68000マシン語/X-BASIC/DōGA・CGA
● Oh!X2周年特別企画「素粒子の声が聞こえる」
● X1/turbo用アクションゲームACTIVE UNIT
LIVE in '89 天空の城ラピュタ/ギャラクシーフォース
THE SOFTOUCH 38万キロの虚空/た～みのる2
全機種共通システム SLANG用リダイレクションライブラリ

1990

1月号
特集1 オペレーティングスタイルの研究
特集2 Cプログラミング応用編
連載 ショートプロぱーてぃ/Z80's Bar
X68000マシン語/C調言語講座/DōGA・CGA
● X1/turbo用シミュレーションゲームSuper Battle
LIVE in '90 さよならを過ぎて/RYDEEN
THE SOFTOUCH レナム/メタルサイト
全機種共通システム WORM KUN/再掲載SLANG
特別付録 X68000 THE SOFTWARE CATALOGUE

2月号
特集 画像圧縮へのアプローチ
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X68000マシン語/C調言語講座/X-BASIC調理実習
● X68000用ゲームプログラムGon Gon
● MZ-700用紙芝居Eyelarth
LIVE in '90 オーダイン/魔女の宅急便
THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリスII
マジックパレット/Mu-1/CYBERNOTE PRO-68K
全機種共通システム 超小型コンパイラTTC++

3月号
特集 MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
C調言語講座/X-BASIC調理実習
● X1/turboシミュレーションCRISIS in Tokyo
LIVE in '90 パワードリフト/スキーム/となりのトトロ
THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター
全機種共通システム 超多機能アセンブラOHM-Z80

4月号
特集 ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
● X1・MZ-2000/2500用RPG The Cave of Dalk
● うわさの68040，ついに登場
LIVE in '90 バーニングフォース(OPMD対応)
THE SOFTOUCH The Fille Professor/HOST PRO-68K
全機種共通システム ファジィコンピュータシミュレータI-MY

5月号
特集 BASICプログラミング
第5回 言わせてくれなくちゃだワ
連載 ショートプロぱーてぃ/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
● 新機種X68000SUPER-HD/EXPERTII/PROII
● ラジコンスティックの製作
LIVE in '90 TURBO OUTRUN
THE SOFTOUCH 天下統一/ポピュラス/Hyperword
全機種共通システム インタプリタ言語STACK

6月号
特集 創刊8周年記念PRO-68K(付録5"2HD)
Oh!Xアンケート結果大分析大会
連載 ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
● X1turbo用コマンドシェルシミュレータ
● ハードウェア工作入門
LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH 三国志II/FAR SIDE MOON/グラナダ
全機種共通システム X68000用S-OS"SWORD"他

ノーマルX1への対応

亀田 雅彦 Kameda Masahiko

お待たせしました。MS-DOS風ファイルユーティリティINTEGRAL X1のノーマルX1対応ルーチンの発表です。同時にマシン語部分のソースリストも公開します。今回のプログラムを使用するには6月号のプログラムが必要ですので注意してください。

今月は，ソースリストの公開と待望のノーマルX1への対応です。これで，X1turboでなくても便利なディスク関係の命令が使えるようになりました。なお，6月号の「INTEGRAL.X」は打ち込み終わったでしょうか？ 6月号のプログラムがいちばんの基本になるので，それがないと今月（来月も）のプログラムは動きません。というか，なにがなんだかわけがわからないことでしょう。悪しからず。

それから，先月号書き忘れてしまったことを書きます。BASICプログラムの「COMMAND.X1」を打ち込むときは，くれぐれも変数名に気をつけてください。1文字違っただけでも，お NEWな変数になってしまいますから。特に，最後のほうのまとめて16進数を代入しているところは，間違えると即暴走につながりますよ（テストするとき，壊されちゃ困るディスクはドライブから抜いておくこと)。どうしても動かないじゃん！ とか，プログラムのバグなんじゃないの？ と思う人は，どこかでX1turboZを使わせてもらって，先月の解説を見て，付属ディスクからプログラムを作り出してください。

また，「INTEGRAL.X1」というのは6月号に掲載された基本プログラムの愛称ですし，KAME-DOSというのは，それらをベースにこれから構築されていくシステムの総称ですのでお見知りおきを。

序論

そもそもこのプログラムは，次の3つのコンセプトに基づいて設計・開発されています。

1) X1において，Human68kやMS-DOSとのファイル&操作性における互換性の確保
2) 初代X1からX1turboZにいたるまで，その能力を最大限引き出すような汎用プログラミング
3) 誰にでも拡張&改造ができるように，BASICでコマンドを作成するフレキシブルな構造

それぞれの特徴は，先月号を読めばわかります。1は，コンバータなどに代表される文字どおりの互換機能[1]。今月は，次の2にあるように，ノーマルX1への対応です（ちなみに来月は3をサポートする予定)。もともとX1全シリーズを念頭においていたので，変更したり新たに打ち直したりするところはわずかで済みました。

また，ハードの都合上サポートできないもの以外は，ほぼX1turbo版と同じ機能を持っています。

それでは，本論に入る前の予備知識として，KAME-DOS動作時のメモリマップを図1に示しておきます。

まず，BASICがメインメモリの半分以上を占拠していて，&HD000からのいちばんおいしい部分にマシン語領域を確保しています。そのあいだには，COMMAND.X1などの制御用BASICプログラムが置かれます。

さて，6月号でも説明したとおり，G-RAMやバンクメモリはRAMディスクとして，またはバッファ領域として確保されます。使うBASIC（や機種）によって多少メモリ構成が違いますが，INTEGRAL.X1自体は不動の4番なのでほとんど変更はありません。

そして，実際にFDCをアクセスしたり，RAMディスクをアクセスしたりするのがマシン語の役割です。読み込んだデータを処理したり，ディレクトリを検索したりするのもマシン語で行います。

BASICプログラムは，ユーザーからの入力処理（実際ただのINPUT命令だったりする）と命令ごとの振り分け，それにその命令のもっともおおまかな処理を引き受けます。もちろん，BASICプログラムやマシン語だけではなんの役にも立ちません。それぞれが密接に連絡しあってひとつの命令をこなしていきます。

今月解説するのは，このマシン語の部分です。外部コマンドを発表して，ユーザー自身がそれを拡張していくには，どうしてもBASICからマシン語を呼び出していくことになります。そのためにも，ソースリストをよく見ておいてください。なお，使用したアセンブラはS-OSのREDAです。

そして,もうひとつはX1用のルーチンです。X1turboの方は，このルーチンは必要ありません（CZ-8FB01上で動かすときも必要ない)。機種別・BASIC別になにが必要でなにがいらないか？ などは，あとでまとめて書いておくので参考にしてください。

図1 メモリマップ

アドレス	内容
0000	BASIC システム領域
	BASIC プログラム領域
D000	ジャンプテーブル1
D060	ワークエリア1
D1A0	プログラム領域1
DF00	階層ディレクトリ用ワークエリア
E000	ジャンプテーブル2
E051	ワークエリア2
E100	固定データ
E200	プログラム領域2
EC40	デバイスドライバ登録テーブル
EC80	パス指定
ECC0	変数保存エリア
ED00	フリーエリア
EE00	汎用バッファ
EF00	フリーエリア
F000	ノーマルX1用ディスクアクセス関係 turboではシステム

ノーマルX1用ルーチン

まず初めにX1用の打ち込み方と，使い方を説明しましょう（くどいようですが，X1 turboでは以下の操作は必要ありません）。リスト1がダンプリストです。ソースリストはX1turbo用と一緒になっています。BASIC(CLEAR&HD000を実行すること）やS-OSから打ち込んでください。終わったら，6月号のマシン語ルーチン（&HD000～&HEDFF）をロードして，

SAVEM "FDC.OBJ", &HD000, &HF151

として，一緒にセーブしておいてください。今後ロードするときは，こちらのほうをロードします。

そうしたら，このルーチンを登録しなければなりません。これまた，6月号の「INTEGRAL.X」というコンフィギュレーションファイルを大幅に変更します。新しく作り直したほうが早いので，後ろにまとめて説明しているところにリストを載せておきます。これを打ち込んで，6月号のINTEGRAL.Xに差し換えてください。

ここまでくれば，登録作業は完了です。あとはX1turbo用と同じように使用できます。違いがあれば，その都度フォローしていきましょう。

現時点でX1でできないことは，2HD，バンクメモリ，F：(MEM1：)デバイス，漢字表示です。階層化ディレクトリはKAME-DOSではサポートしていますが，BASIC側がサポートしていないので注意してください。

それから，「COMMAND.X1」の中の注釈は打ち込まないようにしてください（行は残してコメントを入れない)。X1turboでも，CZ-8FB01を使うときは同じです。実行中，メモリ容量不足で止まることがあります。

データ処理論

では，解説を始めましょう。なお，この連載の中ではプログラムの解説よりも，「いかなるテクを使っているのか」というアルゴリズム関係を中心に，なるべく図と言葉で説明していこうと思います（解析したい方は，ソースを見て)。ですから，他機種のユーザーの方も「X1の記事は飛ばそう！」などといわずに，読み物として「こういう手法もあるんだなあ」ぐらいに思っておくと，将来役に立つかもしれません。

ハードウェア攻略法

DOSとは切っても切れない関係にある，ドライブまわりから攻めてみましょう。ディスクドライブの制御には，必ずFDC（フロッピーディスクコントローラ）という石が，CPUとドライブの仲立ちに入ってきます。こいつに命令を与えることで，間接的にモーターやヘッドを動かしたり，CPUとのあいだでデータの受け渡しを行ったりします。

そして，もうひとつ重要なのがDMAC（ダイレクトメモリアクセスコントローラ）という石です。詳しいことは参考文献にゆずるとして，X1turbo[2]ではこれがCPUの代わりに，FDCとデータのやりとりをしています。なぜ，こんな石を使うのか？　というと，ソフトウェア側でタイミングを取る必要がなくなるからです。たとえば「256バイト読め」という命令をあらかじめ与えておいて，「行け！」と指示を出せば，ドライブ・FDC・DMACのあいだで勝手にデータを読み込んでおいてくれるので，非常に便利です。

その他のデバイス（EMMやMEM）に対する入出力は，すべてCPUがシャコシャコと1バイトずつ転送します。また，画面やキーボードなどのコントロールは，当然のごとくBASICまかせです。

ディスクに関する用語，セクタ・トラック・レコードなどは，囲みで簡単に説明しておきます。詳しく知りたい方は，ユーザーズマニュアルや参考文献を参照してください。

ソフトウェア攻略法

KAME-DOSのシステムは，MS-DOSやCP/Mと同じように，機能別に3つの部分に分かれています。

1）最下層に位置し，ハードウェアに密着してバイト単位で記憶装置からのデータの入出力を担当する部分。レコード単位の処理が多い（リスト3）

2）主に1を利用した，入出力データの管理。あるいは，コマンド実行ルーチンの集合体。3から呼び出されることが多い。1の段階では，どのデータも等価だが，ここでディレクトリ・FAT・ファイルなどの意味を付加する（リスト2）

3）コマンド解釈・実行部分。マンマシンインタフェイスでもある。KAME-DOSではBASIC（6月号の「COMMAND.X1」）

それぞれの対応を図2に示しておきましょう。そして，上記の3つのほかに，「デバイスドライバ」というルーチンもあります。これは，ユーザーがあるデバイスを制御するときに，そのデバイスに対して入出力を行うプログラムのことです。通常，起動時

ディスク用語

トラック：5インチフロッピーディスク上には，バームクーヘンのように数十本の「輪」があって，それをトラックと呼ぶ。2Dであれば，表裏にそれぞれ40本ずつ，計80本あるのが普通である。

セクタ：各機種でフォーマットするときには，そのトラックをまたいくつかの部分に区切る。区切ったひとつ分をセクタと呼ぶ。実際のデータはこのセクタ上に記録されるので，セクタ長とセクタ数がディクス容量を決定する。また，ディスクは1バイトずつデータをアクセスできるわけじゃなくて，このセクタごとにしか読み書きできない。1バイト入出力はソフトの仕事である。

レコード：デバイス上で読み書きできる最小単位のこと。すなわち，フロッピーディスクではセクタとなる。

クラスタ：レコード単位で読み書きできるけれど，それじゃ面倒くさいときにエイヤッと何レコードかをまとめて読み書きするための単位。ファイルは，このクラスタがいくつかつながったものとして管理されている。

ディレクトリ：ファイル名などのファイルごとの情報が詰め込まれているテーブル。

FAT：ディスク上のクラスタ間のつながり具合を管理しておくテーブル。これが破壊されると，ディスクアクセスができなくなる重要な部分。ディレクトリとともに，ディスク上の決まった場所に書き込まれている。

MS-DOSのフォーマットを紹介しておく。

	2HD	2D
片面トラック数	77	40
セクタサイズ（単位はバイト）	1024	512
1トラック当たりのセクタ数	8	9
1クラスタ当たりのセクタ数	1	2

参考：「MS-DOS3.3プログラミング＆リファレンス」平沢正之　日本ソフトバンク

注1）　実際に私も，日本語のコメントをソースリストにつけるために，S-OSファイルをX68000上に移してEDで編集した。もし読者で適当なMS-DOSディスクを持っていない人は，付録ディスクでDIRやTYPEを試してみよう。

注2）　X68000やPC-9801でも，同様の理由でディスクアクセスにはDMAが使われている。アクセス中に音楽が鳴ったりできるのも，CPUがそれだけ楽をしているから。
ノーマルX1では余計なハードは一切ついてないので，CPUがデータの読み書きまで担当している。

にDOSを組み込んだり，切り離したりします。すなわち，DOSは初めから「ディスクをドライブするデバイスドライバ」を組み込んでいるので，ディスクアクセスができることになります。

KAME-DOSでは，ディスク(含むRAMディスク) 以外のドライバは基本的に BASIC での処理になります。もちろん設定を変更してマシン語にすることはできますし，登録を増やすこともできますが，それではKAME-DOSの意味がなくなってしまいます。機会をみてデバイスドライバの拡張も行ってみましょう。囲みにデバイスドライバの登録方法を書いておきます。

さて，外部からマシン語を呼び出す方法ですが，ルーチンごとに必要なワークエリアに値をセットしてコールする方法です。一般的なDOSコールのように，パラメータをレジスタ渡しにする方法ではありません。BASICでの制御なので，このようにしました。コールアドレスおよびワークアドレスはまとめて設定してあるので将来のバージョンアップにも対応可能です。

また，エラーが発生したときには，処理を中断してエラー番号をワークに設定し，BASICの制御下まで戻ってきます。したがって，どこでエラーが起きたとしても必ず「COMMAND.X1」内のエラールーチンにくるようになっています（これを利用すれば，標準エラー出力なんて処理もできるでしょう)。もちろん，BASICのON ERROR GOTOなんぞは一切使用しておりません。

バッファ攻略法

6月号でもちょっと触れておきましたが，ディスクをアクセスする場合には必ずある程度のバッファをメモリ上に取ることになります。あるデバイスからのデータは，まずこのバッファに格納されて，そのあとで指示されたデバイスへと送り込まれます。たとえば，

COPY A：ABC B：

とした場合，まずバッファいっぱいまで，Aドライブから「ABC」というファイルを読み込みます。そして今度はそのデータを，バッファからBドライブへ書き込みます。「ABC」というファイルの容量がバッファより小さければ，1回の「読み書き」で終わりますし，大きければその分だけアクセス回数も増えてしまうでしょう。

「なぜ,こんな面倒なことをするのか？読み書きするメモリ間で直接データをやりとりすればいいじゃん」と，思う人もいるかもしれません。でもそれじゃ，アクセスするデバイスの数だけプログラムが必要になって，能率的ではありません。入力も出力も「すべての道はバッファに通ず」としておけば，どんなデバイスであろうと，そのバッファに対してアクセスするプログラムがあればいいのです。便利。便利。

実は，デバイスドライバという手法も，この考え方からきています。これも図3に示しておきましょう。

ちなみに，X1シリーズのBASICでは,バッファは256バイト固定です。これはとりもなおさず，X1フォーマットの1レコード(1セクタ）に当たります。MS-DOSやHumanでは，CONFIG.SYSのなかで，

BUFF＝20　1024

などとして，設定できるようになっています。1KバイトというのはMS-DOS フォーマットの1レコード（1セクタ）の長さです。つまりバッファの大きさは，1レコードの倍数になっているのが普通なんですね。そして，X1の256バイトというのはとんでもなく小さいので，ディスクアクセスも遅いのです。

KAME-DOSでは最低バッファ容量が4Kバイトになります。これは，X1の1クラスタ，MS-DOSの4レコード分に当たります。これだけのデータをまとめて読み書きするので，理論上BASICの16倍のスピードということになりますか。でも，実際にはRAMディスクアクセス時はバッファが小さいほうが速いので，どちらがいいともいえません。

このように，DOSというのはデータの仲介業者だったりします。自分でも，ディスクをアクセスするルーチンを持っていますが，そういうデバイスドライバ同士のあいだに立ってのデータの受け渡しが本業といえるでしょう。バッファはいわば倉庫であり，デバイスドライバが荷物を運ぶトラックやコンテナで，DOSはそれを仕切っている会社です。それじゃ，人間はなんでしょう？　その会社の社長といえるくらい,DOSを使いこなしていますか？

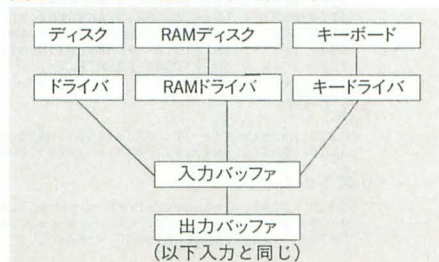



図2　プログラム相関図

KAME-DOS	LIST2	LIST3
COMMAND.X1		
CP/M	BDOS	BIOS
CCP		
MS-DOS	MS-DOS.SYS	IO.SYS
COMMAND.COM		
Human68k	Human.SYS	Human.SYS
COMMAND.X		

図3　バッファ&デバイスドライバ

プログラム分析論

DOSにはいろいろな命令がありますが，マクロ的に見ればそれらの処理がだいたい似ていることがわかります。人間にとっては，BASICファイルとワープロの文書ファイルは全然意味が違いますが，DOSにとってはどちらもただのファイルです。データ(ファイル)の制御という観点から，KAME-DOS の動作概念と個々のルーチンワークを調べてみましょう。

●動作概念

DOSの本質的な動作原理は，入力・処理・出力の3つに大別されます。つまり，ほとんどの命令はこの一連の動作の繰り返しなのです。たとえば，

TYPE A：ABC

という命令は，Aドライブから「ABC」ファイルを（バッファに）入力して,「ファイルの終わりかどうか？　の判断」という処理をして，(バッファから)画面に出力します。入出力はそれぞれのデバイスドライバが担当しますが，「処理」の部分はDOSが行います。エンドコードの検出のほかにも，X1・MS-DOS間の改行コード変換や，ディスク上のディレクトリからディスプレイ形式への変換作業などがあります。

その他，いくつかの注意点があります。入出力するときには，BASICと同じようにファイルOPEN・CLOSEを行います。バッファには入力用（I）と出力用（O）の2つがあって，上記のようなデータ変換を行う場合は，

デバイスドライバの登録法

&HEC40から，ひとつのデバイスごとに6バイト割り振られていて，全部で8個登録できます。そして，そのうち5個は設定ずみなので残りは3個です。6バイトの内訳は，最初の3バイトがデバイス名（NUL，CONなど)，次の1バイトがBASICかマシン語かの設定（0がBASIC，1がマシン語),最後の2バイトはマシン語のジャンプアドレスです（BASICのときは無関係)。4，5，6番目のデバイス名は，見てわかるとおり「$1@」などと架空の名前になっているので,これを書き換えられます。なお，この架空の名前と同じファイル名は指定できません。

「デバイスからデータをIに入力」

「Iから1バイトずつ読んで，変換して，Oに書き込む」

「Oからデータをデバイスへ出力」

という手順を踏むので多少遅くなります。もちろん，変換をしないCOPYなどはIから直接出力するので速くなります。

MS-DOSとX1フォーマットの取り扱いについては，その構造の類似性を利用して，違うところだけをデータ化して対応しています。FATに関しては個別のプログラムで処理しています。またセクタ長の違いも，バッファに入れるときにつじつまを合わせておいて，あとは入力したバイト数でのみ操作する構造にし，ほかのルーチンに余計な負担がかからないようになっています。

それでは，ここから面白そうな個別ルーチンを拾って見てみます。まずはリスト3（ディスクアクセスの基本ルーチン集だよ）から。

●RWREC

1レコード分のデータ入出力。

ディスクをアクセスする際には，必ず呼び出される使用頻度のきわめて高いルーチン。RAMディスクとフロッピーディスクではまったく違うルーチンへ分岐するので，2つに分けて説明します。

A：フロッピーディスク

DMAを使っているので読むのと書くのではそれほど差がなくて，DMA用データとコマンドを変更しているくらい。フォーマットによって違うのはセクタ長だけ。

基本動作として，「バッファのバンクをアクティブにする・ターゲットセクタを算出する・モーターを回し，トラックをシークする・FDCにコマンドを与える・DMAを走らせる・(DMAが勝手に入出力)・モーター停止・バンクをもとへ戻す」の一連の動作をして，そのあいだにエラー判定が入ります。面白いのは，1レコード処理時間内(100分の1秒単位）にこれだけやってしまうことです。もし遅れれば，次のセクタが(ヘッドの下を)通りすぎていて読めなくなり，最高で26倍（X1の2HD）しかも次のセクタでもそれが起こるので，1クラスタ（X1で16セクタ）で最高26×16倍遅くなるのでした。

BASICのディスクアクセスが遅いのは頻繁にこの現象が起こるからのようで，狭いバッファの弊害がここにも出ています。

B：RAMディスク

基本的には「デバイス&バッファバンクの設定・LD命令のループ」の2つだけです。LD命令は，バッファやデバイスによってIN/OUT命令に自己書き換えされます。そして，面倒なのがデバイスとバッファのバンク関係の把握です。表裏バンクの関係であれば，1バイト転送するごとにバンク切り換えをしますし，そうでなければただの読み書きです。たとえば，G-RAM0(E：)にバッファを設定して，G-RAM1(F：)をアクセスすると，遅くなるのはそのためです（これを避けるにはバッファのバッファ!? が必要)。

●CRSRW

ご存じ！　クラスタ読み書き。

入力方法

X1シリーズ各機種でKAME-DOSを使うためには，CONFIGファイル（ファイル名は「INTEGRAL.X」）を書き換える必要があります。ノーマルX1，ノーマルturbo，バンク増設をしたturbo，ノーマルZ，バンク増設（内蔵）のZに分けて，それぞれに標準的なCONFIGファイルを設定しておきます。しかし，「turboでCZ-8FB01を使いたい」などというふうに，ここで紹介していないことをしたいときは，6月号を見て自分で設定してみてください。

ノーマルX1：LIST1（使用BASICはCZ-8FB01 ver 1.0)

プログラムの内容については，6月号を見てください。1250行の，

POKE &HD07F, 0

というのは，「BASICで階層ディレクトリが使えない」という意味のフラグです。

バンク増設をしていないturbo：LIST2（使用BASICはCZ-8FB02)

バンク増設をしたturbo：LIST2＋LIST2.1（使用BASICはCZ-8FB02)

LIST2を，LIST2.1の変更点に従って変更してください。

バンクを増設（内蔵）していないZ：LIST2＋LIST2.2（使用BASICはCZ-8FB02)

LIST2を，LIST2.2に従って変更してください。ちなみに，6月号に掲載したのはこの設定のものです。

バンクを増設（内蔵）しているZ：LIST2＋LIST2.1＋LIST2.2（使用BASICはCZ-8FB02)

LIST2を，LIST2.1およびLIST2.2に従って変更してください。

リスト1

```
1000 '
1010 'INTEGRAL X        CONFIG FILE   LIST 1
1020 '                                 (C) M.Kamdeda 1990
1030 '
1040 WIDTH 80:OPTIONSCREEN 2
1050 DEFINT a-z:CLS 4:INIT:SCREEN
1060 PRINT "KAME DOS  [INTEGRAL.X1]  ver 1.0"
1070 PRINT "   Copyright (C)  1990  M.Kameda"
1080 CLEAR &HD000:LOADM "FDC.OBJ":LOADM "DOS3.OBJ"
1090  m_rwrec=&HE000:MEM$(m_rwrec+1,2)=MKI$(&HF000)
1100  m_wttrc=&HE01E:MEM$(m_wttrc+1,2)=MKI$(&HF003)
1110 v_rscmd=&HE0A6:v_skcmd1=&HE0A7:v_skcmd2=&HE0A8
1120  POKE v_rscmd,2:POKE v_skcmd1,&H1A:POKE v_skcmd2,&H1E
1130 'POKE v_rscmd,0:POKE v_skcmd1,&H18:POKE v_skcmd2,&H1C
1140 v_smacs=&HE0AB:POKE v_smacs,3
1150 v_wait=&HE0AC :POKE v_wait,&H10
1160 v_ctrl=&HD18E:s_path=&HEC80:s_s=&HECC0
1170 s_mac4=s_s   :s_tr4=s_s+26 :s_dn=s_s+30  :s_buff=s_s+31:s_ff=s_s+33
1180 s_bsiz=s_s+35:s_escp=s_s+37:s_iomm=s_s+38:s_eof=s_s+39 :s_eof3=s_s+40
1190 FOR i=s_mac4 TO s_mac4+26:POKE i,3:NEXT:POKE s_tr4,0,0,0,0
1200 INIT "MEM0:":'INIT "MEM1:"
1210 MEM$(s_ff  ,2)=MKI$(&HD000):MEM$(s_buff,2)=MKI$(&HE000)
1220 MEM$(s_bsiz,2)=MKI$(&H1000):POKE s_iomm,1
1230 POKE s_escp,&H12:POKE s_eof,26:POKE s_eof3,26:POKE v_ctrl,32
1240 p$="e:/;w:/;a:/;b:/;"+CHR$(0):MEM$(s_path,LEN(p$))=p$
1250 POKE s_dn,0:POKE &HD07F,0
1260 RUN "COMMAND.X1"
```

リスト2

```
1000 '
1010 'INTEGRAL X        CONFIG FILE    LIST 2
1020 '                                  (C) M.Kamdeda 1990
1030 '
1040 WIDTH 80,25,0,2:KLIST 1:π 1:OPTIONSCREEN 4
1050 DEFINT a-z:CLS 4:INIT:SCREEN
1060 PRINT "KAME DOS  [INTEGRAL.X1]  ver 1.0"
1070 PRINT "   Copyright (C)  1990  M.Kameda"
1080 CLEAR &HD000:LOADM "FDC.OBJ"
1090 'm_rwrec=&HE000:MEM$(m_rwrec+1,2)=MKI$(&HF000)
1100 'm_wttrc=&HE01E:MEM$(m_wttrc+1,2)=MKI$(&HF003)
1110 v_rscmd=&HE0A6:v_skcmd1=&HE0A7:v_skcmd2=&HE0A8
1120  POKE v_rscmd,2:POKE v_skcmd1,&H1A:POKE v_skcmd2,&H1E
1130 'POKE v_rscmd,0:POKE v_skcmd1,&H18:POKE v_skcmd2,&H1C
1140 v_smacs=&HE0AB:POKE v_smacs,3
1150 v_wait=&HE0AC :POKE v_wait,&H10
1160 v_ctrl=&HD18E:s_path=&HEC80:s_s=&HECC0
1170 s_mac4=s_s   :s_tr4=s_s+26 :s_dn=s_s+30  :s_buff=s_s+31:s_ff=s_s+33
1180 s_bsiz=s_s+35:s_escp=s_s+37:s_iomm=s_s+38:s_eof=s_s+39 :s_eof3=s_s+40
1190 FOR i=s_mac4 TO s_mac4+26:POKE i,3:NEXT:POKE s_tr4,0,0,0,0
1200 INIT "MEM0:":INIT "MEM1:"
1210 MEM$(s_ff  ,2)=MKI$(&HD000):MEM$(s_buff,2)=MKI$(&HE000)
1220 MEM$(s_bsiz,2)=MKI$(&H1000):POKE s_iomm,2
1230 POKE s_escp,&H12:POKE s_eof,26:POKE s_eof3,26:POKE v_ctrl,32
1240 p$="w:/;a:/;b:/;"+CHR$(0):MEM$(s_path,LEN(p$))=p$
1250 POKE s_dn,0:POKE &HD07F,1
1260 RUN "COMMAND.X1"
```

リスト2-1

```
1210 MEM$(s_ff  ,2)=MKI$(&H0):MEM$(s_buff,2)=MKI$(&H1000)
1220 MEM$(s_bsiz,2)=MKI$(&H3000):POKE s_iomm,3
```

リスト2-2

```
1120 'POKE v_rscmd,2:POKE v_skcmd1,&H1A:POKE v_skcmd2,&H1E
1130  POKE v_rscmd,0:POKE v_skcmd1,&H18:POKE v_skcmd2,&H1C
1140 v_smacs=&HE0AB:POKE v_smacs,1
```

あらかじめ読み込んでおいたFATを参照しながら，バッファ一杯までデータを読み書きします。レコード長と1クラスタ内のレコード数はフォーマットによって違うので，前もって指定してあるカウンタを使います。RWREC・「クラスタからレコードを計算する」・「FATのつながりを追う（空きFATを追う）」という重要なルーチンを使っているので，要CHECK！　また，1クラスタごとにFATを追っかけるアルゴリズムなので，えんえんと書き込みをした挙句にDISK FULLエラーになる可能性もあります。本当なら実行する前に，きっちりとFATを計算しておくべきなのですが……。

ところで，このルーチンの頭のところでトラックをSEEKしている意味不明の部分があります。いろいろテストしてみると，「2HDモードで連続的に1トラックだけSEEK」するとシークエラーが発生するようです。そのため，ここで2トラック分SEEKして（あとですぐ戻すけど）ごまかしていますが，どうもこれは釈然としませんね。

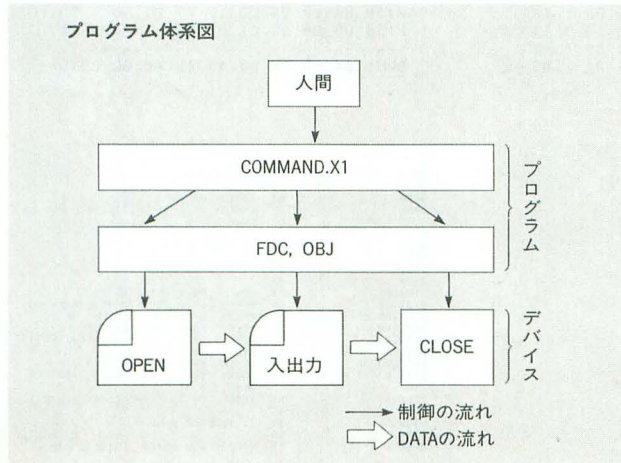
●DEVICE

FATを読み込みつつ，そのディスクのフォーマットを判定しております。読み込み中にエラーが発生したり，FATの内容が正当でなかったりすると，違うフォーマットと判断されます。また，ここで読み込まれたFATが，あとにCRSRWで使われます。

データの扱い

以上でリスト3については終わりにして，リスト2に移ります。ここではハードウェアの接待ではなくて，データを中心とした処理をしています。

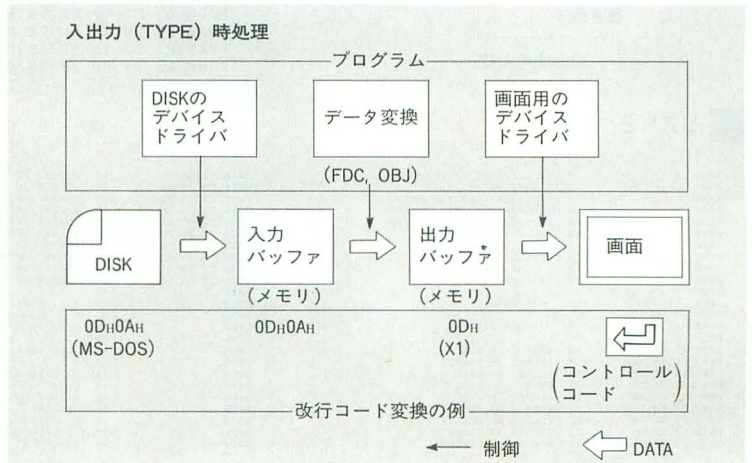
●DIR

その名のとおり，ディレクトリ領域を読み込んで，指定されたファイル名を探し出します。ワイルドカードの取り扱い，サブディレクトリ，ファイル属性などというドロドロした部分があるので，解析はやめたほうがいいでしょう。フォーマット別の対応では，データ化の手法がもっとも効率的に運用されています。「DIR2」は，ほかにも該当するファイル名があるかどうか調べるルーチンです。

●PREOPEN&OPENS

BASICでいうところのOPEN。

デバイスやサブディレクトリ指定の部分の処理と，純粋にファイル名だけの処理にわけています。実際に「ファイルをOPENする」というのは，OPEN番号ごとにワークエリアを設定して，そこにファイル情報をセットすることです。ファイル情報とは，FATアドレス・バッファアドレス・ディレクトリ内での位置・フォーマット・クラスタ・サブディレクトリのクラスタなどと，ファイル名・ファイルサイズなどです。また，ファイル番号はここではコマンド列の順に0，1，2……です。

このルーチンで苦労したのは，ファイル名のセットのところでした。人間の不当ファイル名指定にも果敢に対応していく，というのは非常に大変です。サブディレクトリ名の判定など，あとで見てもわからないくらい汚いところです。それから，大文字/小文字の相互判定方法もCHECKポイントでしょう。ここではファイル名の代わりにデバイス名（PRN，NULなど）の判定もしています。

また，OPEN時には，「DIR」も呼び出されてディレクトリの読み込みとファイル名の検索も行っています。この時点で指定ファイル名が存在しない場合は，エラーが発生します。そして，そのディレクトリ情報をセットしたあとは，読み込んだディレクトリはそのまま捨ててしまいます。

●CLOSE

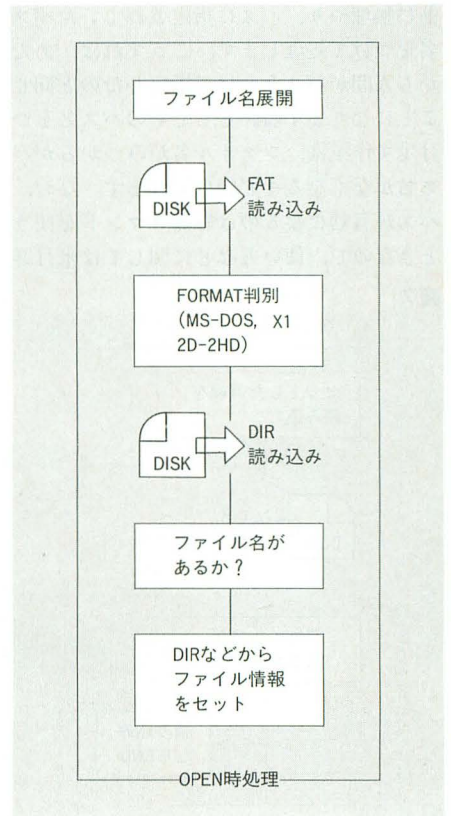
BASICのCLOSEのことです。

実際にはOPENでやったことの逆で，ワークエリアに設定しておいたファイル情報を利用して，ファイルをディスクに登録します。つまり，ディスクアクセスしてる最中はデータをクラスタごとに読み込み（書き込み）しているだけで，メモリ上のFATを使っているだけなのです。そしてひとつのファイルの処理が終わったら，もう一度ディレクトリを読み込んで，ファイル名を捜します。ここではまだクラスタに書き込んだデータを，ファイルとして登録してません（登録するということは，ディレクト

図6

ファイル名展開
DISK　FAT読み込み
FORMAT判別（MS-DOS，X1 2D-2HD）
DISK　DIR読み込み
ファイル名があるか？
DIRなどからファイル情報をセット
OPEN時処理

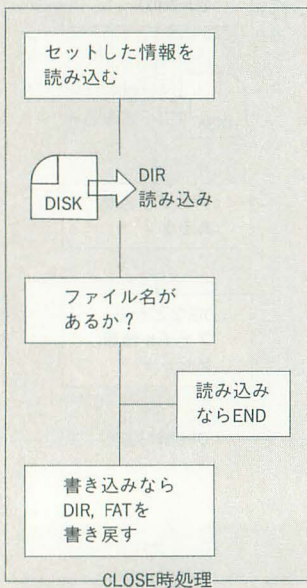
図4

図5

リとFATをディスクに書き戻すことです)。

また，上書きをする場合は，ファイル名を探したときにそれの旧ディレクトリが発見されるはずなので，そのディレクトリとFATを消してその上からを書き込みます。この方法だと，新しいFATを書き込んだあとに古いFATを消去するので，FATにはぼろぼろと「穴」が空いていきますが，まあ，そういう構造なので問題はありません。

それでこのCLOSEのいちばん最後で，上記のような登録をして，やっとひとつのファイル処理が終わるのでした。

●PATH

機能は6月号で説明したとおりです。ファイルをOPENする過程で，ファイル名の前に無理やり，パスに指定されていたパス名をつけて処理します。こうすれば，あたかも人間がパスをつけて指定したのと同じことになりますね。そしてそのパス名をつけ足す作業は，ファイル名がみつかるかパス名がなくなるまで続けられます。なお，パスが有効になるのは外部コマンドを使うときなので，使い方などに関しては来月詳しく説明します。

●MSX1

データ変換を司るルーチン。

前に説明したとおり，入力バッファのデータを出力バッファへ転送するときに，なんらかの変換を加えます。「NOCHECK」は，なんの変換も加えないときのルーチンです。変換にも，エンドコード判定のみ・改行（コントロール）コード変換・その逆の3種類があります。どれにしてもアルゴリズムは簡単で，「1バイト読んで，必要ならひとつ前のデータも参照しながら変換して(あるいはしないで)，転送」の繰り返しです。したがって，これを任意のデータの変換ルーチンとして使うことはできません。

そして，このルーチンのもうひとつ重要な機能は，呼び出したルーチンに対して出力/入力要求をすることです。呼び出したルーチンは，この要求によって入出力バッファのデータを入出力します。また，データの終わりを知らせてくることもあります。いわば，プログラム同士で通信制御を行っているようなものです。変換の際にデータの増減が起こるので，このようなアルゴリズムになりました。

●ノーマル仕様のすごい奴

X1用ルーチンのことです。X1turboでは，漢字変換エリアなどで使われる&HF000以降に，ディスクアクセスルーチンを置きました。「RWREC」ルーチンへのジャンプアドレスを変更しているので，かたちとしては「RWREC」の差し換えです。もうひとつは，「WTTRC」（1トラックフォーマットルーチン）の差し換えになります。この2つを代えるだけで動いてしまうのが,KAME-DOSのいいところでしょう。でも，内容は実は『試験に出るX1』そのまんまです。祝氏に感謝，感謝。

＊　＊　＊

今月も長い解説でしたが，KAME-DOSの概要はわかってもらえましたかねえ？とりあえずわからなくても，興味のあるところの知識が増えれば，それだけ将来役に立つこともあるでしょう。今月の記事は，むしろ他機種ユーザーでも，DOSに興味のある人に読んでほしいところです。DOSの構造を知る手掛かりになるだろうし,「システムプログラムとはこういうもんだ！」というのもわかることでしょう。個々のルーチンのアルゴリズムも，知っておいて損はないと思いますよ。

それでは，来月はもっと使えるDOSにするために．外部コマンドの発表です。とりあえず来月まで，ばははーい。　に，

図7

リスト1

```
F000 C3 06 F0 C3 02 F1 3A 8B : 34
F008 E0 FE 04 D2 C5 E8 C5 D5 : FB
F010 CD 78 E8 38 73 3A 81 EB : 7E
F018 01 FA 0F ED 79 16 FB 1E : 9F
F020 F8 D9 ED 4B 91 E0 D9 3A : 8D
F028 A1 E0 B7 C2 AD F0 F3 CD : 57
F030 B3 E6 21 ED 79 38 03 21 : 7C
F038 02 00 22 A6 F0 3E 80 CD : 45
F040 E6 F0 38 44 CD 9B F0 07 : B1
F048 E6 DF 28 05 CD 31 EA 18 : F2
F050 37 D9 60 69 ED 4B 91 E0 : 82
F058 B7 ED 42 ED 4B 9F E0 B7 : 54
F060 ED 42 D9 28 08 3E 01 32 : A9
F068 8C E0 37 18 1B 2A 9F E0 : 7F
F070 ED 5B 91 E0 19 22 91 E0 : 65
F078 2A 93 E0 23 22 93 E0 01 : 56
--------------------------------
SUM: 09 BA 55 3C 8A 42 26 07 EC0F

F080 FC 0F 3A 8B E0 ED 79 B7 : CD
F088 01 D0 1F 3A A2 E0 ED 79 : 12
F090 01 00 0B 3E 10 ED 79 FB : BB
F098 D1 C1 C9 ED 78 0F D0 0F : AE
F0A0 30 F9 4A ED 78 D9 ED 79 : 17
F0A8 03 D9 4B 18 EE F3 CD B3 : A0
F0B0 E6 21 ED 78 38 03 21 0A : D2
F0B8 00 22 DF F0 22 C0 F0 D9 : 9C
F0C0 ED 78 08 D9 3E A0 CD E6 : D7
F0C8 F0 DA 88 F0 CD D2 F0 C3 : 94
F0D0 47 F0 ED 78 0F D0 0F 30 : BA
F0D8 F9 4A 08 ED 79 D9 03 ED : 7A
F0E0 78 08 D9 4B 18 EC 32 8C : 66
F0E8 EB C5 D5 CD FC E6 D4 6F : 77
F0F0 E7 D1 C1 D8 01 F8 0F 3A : 93
F0F8 8C EB ED 79 3E 07 3D 20 : 7F
--------------------------------
SUM: DB CA 6F F4 B0 44 9B 64 A2C0

F100 FD C9 C5 D5 CD 78 E8 D4 : 61
F108 FC E6 D4 4D E7 38 48 3E : A8
F110 03 32 5D F1 F3 CD B3 E6 : DC
F118 21 ED 78 38 03 21 0A 00 : EC
F120 22 DF F0 22 2F F1 16 FB : 44
F128 1E F8 D9 ED 4B 91 E0 ED : 85
F130 78 08 D9 01 F8 0F 3E F0 : 8F
F138 ED 79 3E 07 3D 20 FD CD : D2
F140 D2 F0 07 E6 DF 28 0C 21 : E3
F148 5D F1 35 C2 26 F1 CD 31 : 5A
F150 EA 18 04 CD 1F E7 B7 CD : 5D
F158 D9 E6 D1 C1 C9 00       : 1A
--------------------------------
SUM: B4 05 5F 98 46 4F AE BC C67D
```

リスト2

```
0000                 1  ;///////////////////////////
0000                 2  ; INTEGRAL X  No.1
0000                 3  ;  (KAME-DOS) for REDA
0000                 4  ; COPYRIGHT 1990 M.KAMEDA
0000                 5  ;///////////////////////////
0000                 6  ;
0000                 7
0000                 8  ;///////////////
0000                 9  ; JUMP TABLE
0000                10  ;///////////////
0000                11
D000                12   ORG $D000
D000                13
D000                14  ;/ JUMP TABLE /
D000                15  ;
D000 C3 87 D7       16  #DIR   JP DIR
D003 C3 A5 D7       17  #DIRWT JP DIRWT
D006 C3 2F D8       18  #DIR2  JP DIR2
D009 C3 43 DA       19  #MSX1  JP MSX1
D00C C3 7D D5       20  #ICPO  JP ICPO
D00F C3 6D DD       21  #TRS   JP TRS256
D012 C3 0C DA       22  #DLDIR JP DELDIR
D015 C3 D3 D6       23  #CLOSE JP CLOSE
D018 C3 3C D7       24  #CLOS2 JP CLOSE2
D01B C3 8C D7       25  #DIR3  JP DIR3
D01E C3 20 DA       26  #DFREE JP DFRE
D021 C3 F3 D3       27  #PREOP JP PREOPEN
D024 C3 CA D4       28  #OPENS JP OPENS
D027 C3 BB D5       29  #DVDRV JP DEVDRV
D02A C3 28 DC       30  #DIRSB JP DIRSUB
D02D C3 85 DD       31  #SETDN JP SETDN
D030 C3 4A D2       32  #TRAN  JP TRAN
D033 C3 A0 D1       33  #ALTA  JP ALTA
D036 C3 D3 D1       34  #TRANR JP TRANRT
D039                35
D039                36  ;//////////////
D039                37  ;    PROGRAM
D039                38  ;//////////////
D039                39
D060                40   ORG $D060
D060                41
D060                42  ;/ OPTION WORK /
D060                43  ;
D060                44  #ALPHA
D060                45   DS 26
D07A                46
D07A                47  ;/ OPEN WORK /
D07A                48  ;
D080                49   ORG $D080
D080                50  ;
D080                51  #ZOKU1 DS 4
D084                52  #FNAM1 DS 46*4 ;+35 F,M$,D,M,C,DI
                        RN,C2
D13C                53
D13C                54  ;/ WORK エリア 1 /
D13C                55  ;
D180                56   ORG $D180
D180                57  ;
D180 01             58  #OD    DB 1 ;FILE 番号
D181 00             59  #OP    DB 0 ;OPEN時 機能指示FLUG
D182 00 00          60  #FCRS  DW 0 ;空クラスター数
D184 00             61  #YEN   DB 0 ;文字列の大きさ
D185 00             62  #CDF   DB 0 ;コマンドがCDか?FLUG
D186 00 00          63  #FSZL  DW 0 ;
D188 00 00          64  #FSZH  DW 0 ;FILE-SIZE 4バイト
D18A 07 07 07 07    65  #DDRV  DB 7,7,7,7 ;DEVICE-DRIVER
                        撰択
D18E 20             66  #CTRL  DB 32 ;CTRL-CODE の上限
D18F                67
D18F                68  ;/ 階層DIR WORK /
D18F                69  ;
DF00 P              70  CSDIRN EQU $DF00    ;階層の深さ
```

▶“この木なんの木”みたいな音楽をどしどし載せてほしい。たとえば，およげたいやきくんとか？　パーマンの歌とかランバダとか……。ラジオ体操第1と第2があったなぁ。もちろんおっさんのPCMを入れるとか……。　秋本　知彦 (18) 山口県

```
DF1A P              71  CSDIR  EQU $DF00+26 ;そのCLUSTER
D18F                72
D18F                73  ;////////////
D18F                74  ;/ PROGRAM /
D18F                75  ;////////////
D18F                76
D1A0                77   ORG $D1A0
D1A0                78
D1A0                79  ;/ OPTION 処理 /
D1A0                80  ;
D1A0                81  ALTA ;IN (DE),B
D1A0 1A             82   LD A,(DE)
D1A1 CD B1 D1       83   CALL ALPHA?    ;大文字?
D1A4 DC F7 D8       84   CALL C,OKCHG   ;IF 違う CALL 変換
D1A7 CD B1 D1       85   CALL ALPHA?    ;大文字?
D1AA D4 B8 D1       86   CALL NC,NUMB?  ;数字?
D1AD 13             87   INC DE         ;次の1文字へ
D1AE 10 F0          88   DJNZ ALTA
D1B0 C9             89   RET
D1B1                90
D1B1                91  ALPHA? ;アルファベット大文字か?
D1B1 FE 41          92   CP 'A'
D1B3 D8             93   RET C
D1B4 FE 5B          94   CP 'Z'+1
D1B6 3F             95   CCF
D1B7 C9             96   RET
D1B8                97
D1B8                98  NUMB? ;数字か? & WORKへSET
D1B8 C5             99   PUSH BC
D1B9 D6 41         100   SUB 'A'        ;
D1BB 4F            101   LD C,A         ;
D1BC 06 00         102   LD B,0         ;
D1BE 21 60 D0      103   LD HL,#ALPHA ;WORK エリア
D1C1 09            104   ADD HL,BC      ;HL=WORK+(DE)-'A'
D1C2 13            105   INC DE
D1C3 1A            106   LD A,(DE)
D1C4 1B            107   DEC DE
D1C5 FE 30         108   CP '0'         ;
D1C7 38 07         109   JR C,NUMBRT    ;
D1C9 FE 3A         110   CP '9'+1       ;
D1CB 30 03         111   JR NC,NUMBRT ;数字なら それをSET
D1CD D6 31         112   SUB '0'+1      ;
D1CF 77            113   LD (HL),A      ;違うなら 1をSET
D1D0               114  NUMBRT
D1D0 34            115   INC (HL)
D1D1 C1            116   POP BC
D1D2 C9            117   RET
D1D3               118
D1D3               119  ;/ PATHに従ってFILEをさがす /
D1D3               120  ;
D1D3               121  TRANRT ;IN (DE),B
D1D3 AF            122   XOR A
D1D4 32 80 D1      123   LD (#OD),A
D1D7 32 81 D1      124   LD (#OP),A
D1DA CD 4A D2      125   CALL TRAN
D1DD D8            126   RET C
D1DE 3A 97 E0      127   LD A,(#MAC)
D1E1 FE 02         128   CP 2     ;MS-DOS
D1E3 28 5E         129   JR Z,NTRBOT
D1E5 FE 04         130   CP 4     ;MS-DOS
D1E7 28 5A         131   JR Z,NTRBOT
D1E9 3A 8B E0      132   LD A,(#DN)
D1EC FE 04         133   CP 4
D1EE 30 0E         134   JR NC,SKRBOT ;IF ドライブ=>E:
D1F0 C5            135    PUSH BC
D1F1 D5            136    PUSH DE
D1F2 87            137    ADD A,A
D1F3 6F            138    LD L,A
D1F4 26 00         139    LD H,0
D1F6 11 FA DD      140    LD DE,DISK0 ;DATA エリア
D1F9 01 02 00      141    LD BC,2      ;DATAの長さ
D1FC 18 1D         142    JR SKRBOT2
D1FE               143  SKRBOT
D1FE FE 06         144   CP 6         ;
D200 38 06         145   JR C,OKRBOT ;
D202 FE 16         146   CP 22        ;
D204 38 3D         147   JR C,NTRBOT ;G:～V:はダメ
D206 D6 10         148    SUB 22-6
D208               149  OKRBOT
D208 C5            150   PUSH BC
D209 D5            151   PUSH DE
D20A 21 05 00      152   LD HL,5      ;
D20D D6 04         153   SUB 4        ;A=A-4
D20F 5F            154   LD E,A       ;
D210 16 00         155   LD D,0       ;
D212 CD 27 E0      156   CALL #MLTHD ;HL=5*A
D215 11 02 DE      157   LD DE,MEM0  ;DATA エリア
D218 01 05 00      158   LD BC,5     ;DATAの長さ
D21B               159  SKRBOT2
D21B 19            160   ADD HL,DE
D21C 11 81 EE      161   LD DE,P256+$81
D21F ED B0         162   LDIR    ;ドライブ名SET
D221 21 01 EE      163   LD HL,P256+1
D224 3A 00 EE      164   LD A,(P256)
D227 B7            165   OR A
D228 28 05         166   JR Z,SKRBBB
D22A 4F            167   LD C,A
D22B 06 00         168   LD B,0
D22D ED B0         169   LDIR     ;階層DIR名 SET
D22F               170  SKRBBB
D22F E1            171   POP HL
D230 C1            172   POP BC
D231 48            173   LD C,B
D232 06 00         174   LD B,0
D234 ED B0         175   LDIR     ;ファイル名 SET
D236 21 81 EE      176   LD HL,P256+$81
D239 EB            177   EX DE,HL
D23A B7            178   OR A
D23B ED 52         179   SBC HL,DE
D23D 7D            180   LD A,L    ;A=文字の長さ
D23E 32 80 EE      181   LD (P256+$80),A
D241 B7            182   OR A
D242 C9            183   RET
D243               184
D243               185  NTRBOT
D243 3E 0C         186   LD A,12
D245 32 8C E0      187   LD (#STOP),A
D248 37            188   SCF
D249 C9            189   RET
D24A               190
D24A               191  ;/ FILEがみつかるまで
D24A               192  ;   MAKE FILE-NAME & OPEN /
D24A               193  TRAN ;IN (DE),B
D24A 21 80 EC      194   LD HL,PATH ;
D24D 22 A9 E0      195   LD (#BPATH),HL ;PATH 初期化
D250               196  TRANLP
D250 C5            197   PUSH BC
D251 D5            198   PUSH DE
D252 CD 48 E0      199   CALL #SPATH ;FILE-NAME SET
D255 D1            200   POP DE
D256 C1            201   POP BC
D257 D8            202   RET C
D258 C5            203   PUSH BC
D259 D5            204   PUSH DE
D25A 3A 80 EE      205   LD A,(P256+$80)
D25D 47            206   LD B,A       ;B=LENGTH
D25E 11 81 EE      207   LD DE,P256+$81
D261 CD 69 D2      208   CALL MCPATH ;(DE)=FILE-NAME
D264 D1            209   POP DE
D265 C1            210   POP BC
D266 38 E8         211   JR C,TRANLP
D268 C9            212   RET
D269               213
D269               214  ;(PATHに従いFILEをOPEN)
D269               215  ;
D269               216  MCPATH ;IN (DE),B,OD,OP
D269 C5            217   PUSH BC
D26A D5            218   PUSH DE
D26B CD 21 D0      219   CALL #PREOP ;ドライブ 階層 OPEN
D26E D1            220   POP DE
D26F C1            221   POP BC
D270 D8            222   RET C
D271 3A 84 D1      223   LD A,(#YEN) ;フルパスFILE-NAME
D274 4F            224   LD C,A       ;から、ドライブと
D275 78            225   LD A,B       ;階層の部分を取り
D276 91            226   SUB C        ;除く。
D277 EB            227   EX DE,HL     ;
D278 5F            228   LD E,A       ;
D279 16 00         229   LD D,0       ;
D27B 19            230   ADD HL,DE    ;DE=FILE-NAMEのみの
D27C EB            231   EX DE,HL     ;先頭 ADR
D27D 41            232   LD B,C       ;B=その長さ
D27E 3E 01         233   LD A,1
D280 32 65 E0      234   LD (#SBDR),A
D283 C5            235   PUSH BC
D284 D5            236   PUSH DE
D285 CD 24 D0      237   CALL #OPENS ;FILE-NAME OPEN
D288 D1            238   POP DE
D289 C1            239   POP BC
D28A C9            240   RET
D28B               241
D28B               242  ;(階層DIRを降りる)
D28B               243  ;
D28B               244  KAISO ;IN (DE)
D28B AF            245   XOR A
D28C 21 00 EE      246   LD HL,P256
D28F 77            247   LD (HL),A
D290 23            248   INC HL
D291 22 F8 DD      249   LD (DDUSH),HL
D294 3E 02         250   LD A,2
D296 32 81 D1      251   LD (#OP),A   ; .*
D299 32 65 E0      252   LD (#SBDR),A ;DIR のみ
D29C 78            253   LD A,B
D29D B7            254   OR A
D29E 28 05         255   JR Z,NLEN0
D2A0 CD EF D3      256   CALL HKIGO
D2A3 28 05         257   JR Z,NCRENT
D2A5               258  NLEN0
D2A5 CD 49 D3      259    CALL SETCRD ;SET CURRENT DIR
D2A8 18 09         260    JR KAISO2
D2AA               261  NCRENT
D2AA AF            262   XOR A       ;ROOT
D2AB 32 51 E0      263   LD (#DIRN),A
D2AE CD 93 D3      264   CALL CHGDIR ;変更を登録
D2B1               265  KAISOLP
D2B1 13            266   INC DE
D2B2 05            267   DEC B
D2B3               268  KAISO2
D2B3 CD C5 D3      269   CALL TENTEN ;CD ..
D2B6 38 50         270   JR C,ERR8
D2B8 78            271   LD A,B       ;
D2B9 B7            272   OR A       ;
D2BA 28 5A         273   JR Z,NOYEN2 ;LENGTH END
D2BC D5            274   PUSH DE
D2BD CD 77 D3      275   CALL FINDYEN
D2C0 6F            276   LD L,A
D2C1 D1            277   POP DE
D2C2 0E 00         278   LD C,0
D2C4 30 05         279   JR NC,KAISOSK
D2C6 7C            280    LD A,H ;
D2C7 B7            281    OR A  ;DIR-NAMEじゃない
D2C8 20 4C         282    JR NZ,NOYEN2
D2CA 0C            283    INC C  ;'/'がないよFLUG
D2CB               284  KAISOSK
D2CB C5            285    PUSH BC
D2CC 45            286    LD B,L ;LENGTH
D2CD D5            287    PUSH DE
D2CE CD 21 D6      288     CALL MOTHR ;FILE-NAME 展開
D2D1 2A A4 E0      289     LD HL,(#BADR)
D2D4 22 52 E0      290     LD (#MORF),HL
D2D7 D4 00 D0      291     CALL NC,#DIR ;DIR READ&FIND
D2DA D1            292    POP DE
D2DB C1            293    POP BC
D2DC 38 20         294     JR C,NOYEN ;IF 概当なし JR
D2DE CD 20 D3      295    CALL MKPTH
D2E1 21 51 E0      296    LD HL,#DIRN ;
D2E4 34            297    INC (HL) ;
D2E5 3E 04         298     LD A,4 ;
D2E7 BE            299     CP (HL) ;
D2E8 38 1E         300     JR C,ERR8 ;4階層以上はダメ
D2EA 2A 7C E0      301    LD HL,(#ECRS)
D2ED 22 95 E0      302    LD (#CRS),HL
D2F0 CD 93 D3      303    CALL CHGDIR
D2F3 CD 77 D3      304    CALL FINDYEN
D2F6 38 1B         305     JR C,NOYEN3
D2F8 4F            306     LD C,A
D2F9 78            307     LD A,B
D2FA 91            308     SUB C
D2FB 47            309     LD B,A
D2FC 18 B3         310    JR KAISOLP ;NEXT STEP
D2FE               311
D2FE               312  NOYEN ;終了処理
D2FE 3A 85 D1      313   LD A,(#CDF)
D301 B7            314   OR A
D302 20 04         315   JR NZ,ERR8
D304 79            316   LD A,C
D305 B7            317   OR A
D306 20 0E         318   JR NZ,NOYEN2
D308               319  ERR8
D308 78            320    LD A,B
D309 32 84 D1      321    LD (#YEN),A
D30C 3E 08         322    LD A,8
D30E 32 8C E0      323    LD (#STOP),A
D311 37            324    SCF
D312 C9            325    RET
D313               326  NOYEN3
D313 AF            327   XOR A
D314 18 05         328   JR NOYENSK
D316               329  NOYEN2
D316 AF            330   XOR A
D317 32 8C E0      331   LD (#STOP),A
D31A 78            332   LD A,B
D31B               333  NOYENSK
D31B 32 84 D1      334   LD (#YEN),A
D31E B7            335   OR A
D31F C9            336   RET
D320               337
D320               338  ;(SUBDIR-NAMEの
D320               339  ;  うしろのスペースを飛ばす)
D320               340  MKPTH
D320 D9            341   EXX
D321 01 0D 00      342   LD BC,13
D324 ED 5B F8 DD   343   LD DE,(DDUSH)
D328 21 68 E0      344   LD HL,#FNAM
D32B ED B0         345   LDIR ;SUB-DIR-NAME SET
D32D               346  MKPTHLP
D32D 1B            347   DEC DE
D32E 1A            348   LD A,(DE)
D32F FE 20         349   CP ' '
D331 28 FA         350   JR Z,MKPTHLP
D333 13            351   INC DE      ;
D334 3E 2F         352   LD A,'/'    ;
D336 12            353   LD (DE),A ;'/'をつける
D337 13            354   INC DE      ;
D338 ED 53 F8 DD   355   LD (DDUSH),DE
D33C 21 01 EE      356   LD HL,P256+1
D33F EB            357   EX DE,HL
D340 B7            358   OR A
D341 ED 52         359   SBC HL,DE
D343 7D            360   LD A,L
D344 32 00 EE      361   LD (P256),A
D347 D9            362   EXX
D348 C9            363   RET
D349               364
D349               365  ;(現在のDIRの
D349               366  ;  クラスターと深さをSET)
D349               367  SETCRD
D349 D9            368   EXX
D34A 21 00 DF      369   LD HL,CSDIRN ;DATA エリア
D34D CD B8 DD      370   CALL DNST2
D350 7E            371   LD A,(HL)
D351               372  SETCRD2
D351 32 51 E0      373   LD (#DIRN),A ;深さ
D354 3D            374   DEC A
D355 87            375   ADD A,A
D356 4F            376   LD C,A
D357 06 00         377   LD B,0
D359 3A 8B E0      378   LD A,(#DN)
D35C 5F            379   LD E,A
D35D 16 00         380   LD D,0
D35F 21 08 00      381   LD HL,8
D362 CD 27 E0      382   CALL #MLTHD
D365 11 1A DF      383   LD DE,CSDIR  ;DATA エリア2
D368 19            384   ADD HL,DE
D369 09            385   ADD HL,BC
D36A 5E            386   LD E,(HL)
D36B 23            387   INC HL
D36C 56            388   LD D,(HL) ;クラスター
D36D ED 53 95 E0   389   LD (#CRS),DE
D371 ED 53 7C E0   390   LD (#ECRS),DE
D375 D9            391   EXX
D376 C9            392   RET
D377               393
D377               394  ;(次の'/'をさがす)
D377               395  ;
D377               396  FINDYEN ;OUT A
D377 C5            397   PUSH BC
D378 0E 00         398   LD C,0
D37A 61            399   LD H,C
D37B 78            400   LD A,B ;LENGTH
D37C B7            401   OR A
D37D 28 0E         402   JR Z,YENRTC
D37F               403  FINDYLP
D37F CD EF D3      404   CALL HKIGO
D382 28 0A         405   JR Z,YENRT
D384 FE 2A         406    CP '*' ;NO DIR-NAME
D386 CC 91 D3      407    CALL Z,FINDWSK
D389 13            408   INC DE
D38A 0C            409   INC C
D38B 10 F2         410   DJNZ FINDYLP
D38D               411  YENRTC
D38D 37            412   SCF
D38E               413  YENRT
D38E 79            414   LD A,C
D38F C1            415   POP BC
D390 C9            416   RET
D391               417
D391               418  FINDWSK
D391 24            419   INC H
D392 C9            420   RET
D393               421
D393               422  CHGDIR
D393 3A 85 D1      423    LD A,(#CDF) ;登録FLUG
D396 B7            424    OR A
D397 C8            425    RET Z
D398 D9            426   EXX
D399 21 00 DF      427   LD HL,CSDIRN
D39C CD B8 DD      428   CALL DNST2
D39F 3A 51 E0      429   LD A,(#DIRN) ;階層の深さ
D3A2 77            430   LD (HL),A     ;
D3A3 D6 01         431   SUB 1 ;ROOT
D3A5 38 1C         432    JR C,CHGDRT
D3A7 87            433   ADD A,A
D3A8 4F            434   LD C,A
D3A9 06 00         435   LD B,0
D3AB 3A 8B E0      436   LD A,(#DN)
D3AE 5F            437   LD E,A
D3AF 16 00         438   LD D,0
D3B1 21 08 00      439   LD HL,8
D3B4 CD 27 E0      440   CALL #MLTHD
D3B7 11 1A DF      441   LD DE,CSDIR
D3BA 19            442   ADD HL,DE
D3BB 09            443   ADD HL,BC
D3BC ED 5B 7C E0   444   LD DE,(#ECRS)
D3C0 73            445   LD (HL),E
D3C1 23            446   INC HL
D3C2 72            447   LD (HL),D ;クラスター
D3C3               448  CHGDRT
D3C3 D9            449   EXX
D3C4 C9            450   RET
D3C5               451
D3C5               452  ;(CD .. の処理)
D3C5               453  ;
D3C5               454  TENTEN
D3C5 1A            455   LD A,(DE)
D3C6 FE 2E         456   CP '.'
D3C8 20 23         457   JR NZ,TENTRT
D3CA 13            458   INC DE
D3CB 1A            459   LD A,(DE)
D3CC 1B            460   DEC DE
D3CD FE 2E         461   CP '.'
D3CF 20 1C         462   JR NZ,TENTRT
D3D1 3A 51 E0      463   LD A,(#DIRN)
D3D4 D6 01         464   SUB 1 ;=DEC A
D3D6 D8            465   RET C ;IF ROOT RET
D3D7 D9            466   EXX
D3D8 CD 51 D3      467   CALL SETCRD2
D3DB CD 93 D3      468   CALL CHGDIR
D3DE 13            469   INC DE
D3DF 13            470   INC DE
D3E0 05            471   DEC B
D3E1 05            472   DEC B
D3E2 28 09         473   JR Z,TENTRT
D3E4 CD EF D3      474    CALL HKIGO
D3E7 28 02         475    JR Z,TENTRT2
D3E9 37            476     SCF
D3EA C9            477     RET
D3EB               478  TENTRT2
D3EB 13            479    INC DE
D3EC 05            480    DEC B
D3ED               481  TENTRT
D3ED B7            482   OR A
D3EE C9            483   RET
D3EF               484
D3EF               485  ;
D3EF               486  HKIGO
D3EF 1A            487   LD A,(DE)
D3F0 FE 2F         488   CP '/'
D3F2 C9            489   RET
D3F3               490
D3F3               491  ;/ OPEN 処理 1 /
D3F3               492  ;
D3F3               493  PREOPEN ;IN (DE),OD
D3F3 C5            494   PUSH BC
D3F4 CD 94 D4      495   CALL HDRV ;DRIVER 判定
D3F7 21 8A D1      496   LD HL,#DDRV
D3FA CD 6C D7      497   CALL ODST
D3FD C1            498   POP BC
D3FE 7E            499   LD A,(HL)
D3FF FE 07         500   CP 7
D401 C0            501   RET NZ ;IF DISK以外 RET
D402 CD 2D D0      502   CALL #SETDN
D405 D8            503   RET C
D406 C5            504   PUSH BC
D407 D5            505   PUSH DE
D408 CD 76 D7      506   CALL ODST2
D40B E5            507   PUSH HL ;OPEN情報 WORK
D40C DD E1         508   POP IX  ;=LD IX,HL
D40E 3A 80 D1      509   LD A,(#OD)
D411 D6 01         510   SUB 1  ;=DEC A
D413 CE 00         511   ADC A,0      ;OPEN番号
D415 6F            512   LD L,A       ;によって
D416 26 00         513   LD H,0       ;使うメモ
D418 11 00 08      514   LD DE,$800   ;リーを$8
D41B CD 27 E0      515   CALL #MLTHD ;00ずつ移
D41E ED 5B E1 EC   516   LD DE,(&FF) ;動する
D422 19            517   ADD HL,DE    ;
D423 DD 75 23      518   LD (IX+35),L
D426 DD 74 24      519   LD (IX+36),H
D429 DD E5         520   PUSH IX
```

▶6月号を読んだ後、5月号を引っぱり出して読んでたんです。そしたらなんと、はみだしに私の名前があるじゃないですか。感動もんですねぇ。ん、まてよ、これで私が浪人したことが日本全国に知れわたったということか……。　★ 敬一 (18) 神奈川県

```
D42B D5          521   PUSH DE
D42C 22 8D E0    522   LD (#FF),HL ;ADR SET
D42F 3A 97 E0    523   LD A,(#MAC)
D432 32 F7 DD    524   LD (SMAC),A
D435 CD 0C E0    525   CALL #DEVIC ;自動判定
D438 E1          526   POP HL
D439 DD E1       527   POP IX
D43B 22 8D E0    528   LD (#FF),HL
D43E D1          529   POP DE
D43F C1          530   POP BC
D440 D8          531   RET C
D441 CD 83 D4    532   CALL CSDBM
D444 3A 8B E0    533   LD A,(#DN)     ;OPENした
D447 DD 77 27    534   LD (IX+39),A   ;情報をSE
D44A 2A A4 E0    535   LD HL,(#BADR)  ;T
D44D DD 75 25    536   LD (IX+37),L   ;
D450 DD 74 26    537   LD (IX+38),H   ;
D453 3A 97 E0    538   LD A,(#MAC)    ;
D456 DD 77 28    539   LD (IX+40),A   ;
D459 DD E5       540   PUSH IX
D45B 3A 81 D1    541    LD A,(#OP)
D45E 6F          542    LD L,A
D45F 3A 65 E0    543    LD A,(#SBDR)
D462 67          544    LD H,A
D463 E5          545    PUSH HL
D464 CD 8B D2    546    CALL KAISO ;階層をCHECK
D467 E1          547    POP HL
D468 7C          548    LD A,H
D469 32 65 E0    549    LD (#SBDR),A
D46C 7D          550    LD A,L
D46D 32 81 D1    551    LD (#OP),A
D470 DD E1       552   POP IX
D472 D8          553   RET C
D473 3A 51 E0    554   LD A,(#DIRN)
D476 DD 77 2B    555   LD (IX+43),A
D479 2A 7C E0    556   LD HL,(#ECRS)
D47C DD 75 2C    557   LD (IX+44),L
D47F DD 74 2D    558   LD (IX+45),H
D482 C9          559   RET
D483             560
D483             561  ;(ROOT-DIRへもどす)
D483             562  ;
D483             563  CSDBM
D483 3A F7 DD    564   LD A,(SMAC)
D486 21 97 E0    565   LD HL,#MAC
D489 BE          566   CP (HL)
D48A C8          567   RET Z ;IF 同じFORMAT RET
D48B 21 00 DF    568   LD HL,CSDIRN
D48E CD B8 DD    569   CALL DNST2
D491 AF          570   XOR A ;ROOTへ
D492 77          571   LD (HL),A
D493 C9          572   RET
D494             573
D494             574  ;(DEVICE-DRIVER 振分け)
D494             575  ;
D494             576  HDRV
D494 78          577   LD A,B
D495 FE 03       578   CP 3
D497 C0          579   RET NZ IF 3文字以外 RET
D498 AF          580   XOR A
D499 32 84 D1    581   LD (#YEN),A
D49C 21 40 EC    582   LD HL,SRDR ;TABLE
D49F 0E 00       583   LD C,0
D4A1             584  HDRVLP2
D4A1 D5          585   PUSH DE
D4A2 E5          586   PUSH HL
D4A3 06 03       587   LD B,3 ;3文字
D4A5             588  HDRVLP
D4A5 1A          589   LD A,(DE)
D4A6 BE          590   CP (HL)
D4A7 28 06       591   JR Z,HDRVSSK
D4A9 CD F7 D8    592    CALL OKCHG
D4AC BE          593    CP (HL)
D4AD 20 0F       594    JR NZ,HDRVSK
D4AF             595  HDRVSSK
D4AF 13          596   INC DE
D4B0 23          597   INC HL
D4B1 10 F2       598   DJNZ HDRVLP
D4B3 E1          599   POP HL          ;DRIVER 指定
D4B4 21 8A D1    600   LD HL,#DDRV     ;WORK TABLE
D4B7 51          601   LD D,C          ;識別CODE
D4B8 CD 6C D7    602   CALL ODST
D4BB 72          603   LD (HL),D
D4BC D1          604   POP DE
D4BD C9          605   RET
D4BE             606  HDRVSK
D4BE 0C          607   INC C
D4BF E1          608   POP HL
D4C0 11 06 00    609   LD DE,6 ;6 SKIP
D4C3 19          610   ADD HL,DE
D4C4 D1          611   POP DE
D4C5 7E          612   LD A,(HL)
D4C6 B7          613   OR A
D4C7 20 D8       614   JR NZ,HDRVLP2
D4C9 C9          615   RET
D4CA             616
D4CA             617  ;/ OPEN 処理 2 /
D4CA             618  ;
D4CA             619  OPENS ;IN (DE)
D4CA CD 21 D6    620   CALL MOTHR ;NAME SET
D4CD D8          621   RET C
D4CE CD 76 D7    622   CALL ODST2
D4D1 E5          623   PUSH HL
D4D2 DD E1       624   POP IX
D4D4 DD 6E 23    625   LD L,(IX+35) ;以下
D4D7 DD 66 24    626   LD H,(IX+36) ;情報設定
D4DA 22 8D E0    627   LD (#FF),HL
D4DD DD 6E 25    628   LD L,(IX+37)
D4E0 DD 66 26    629   LD H,(IX+38)
D4E3 22 52 E0    630   LD (#MORF),HL
D4E6 DD 7E 27    631   LD A,(IX+39)
D4E9 32 8B E0    632   LD (#DN),A
D4EC DD 7E 28    633   LD A,(IX+40)
D4EF 32 97 E0    634   LD (#MAC),A
D4F2 CD 1B E0    635   CALL #VAR
D4F5 CD 1E D0    636   CALL #DFREE
D4F8 2A 95 E0    637   LD HL,(#CRS)
D4FB 22 82 D1    638   LD (#FCRS),HL
D4FE DD 7E 2B    639   LD A,(IX+43)
D501 32 51 E0    640   LD (#DIRN),A
D504 DD 6E 2C    641   LD L,(IX+44)
D507 DD 66 2D    642   LD H,(IX+45)
D50A 22 95 E0    643   LD (#CRS),HL
D50D DD E5       644   PUSH IX
D50F CD 44 D5    645   CALL OPN2
D512 DD E1       646   POP IX
D514 D8          647   RET C
D515 CD 76 D7    648    CALL ODST2
D518 EB          649    EX DE,HL
D519 21 68 E0    650    LD HL,#FNAM
D51C 01 23 00    651    LD BC,35
D51F ED B0       652    LDIR
D521 2A 7C E0    653   LD HL,(#ECRS)
D524 DD 75 29    654   LD (IX+41),L
D527 DD 74 2A    655   LD (IX+42),H
D52A DD 75 14    656   LD (IX+20),L
D52D DD 74 15    657   LD (IX+21),H
D530 2A 52 E0    658   LD HL,(#MORF)
D533 DD 75 25    659   LD (IX+37),L
D536 DD 74 26    660   LD (IX+38),H
D539 21 80 D0    661   LD HL,#ZOKU1
D53C CD 6C D7    662   CALL ODST
D53F 3A A3 E0    663   LD A,(#ZOKU)
D542 77          664   LD (HL),A
D543 C9          665   RET
D544             666
D544             667  OPN2
D544 3A 81 D1    668   LD A,(#OP)
D547 FE 03       669   CP 3
D549 DA 00 D0    670   JP C,#DIR
D54C 28 05       671   JR Z,OPNSK2
D54E CD 00 D0    672   CALL #DIR
D551 18 03       673   JR OPNSK3
D553             674  OPNSK2
D553 CD 1B D0    675   CALL #DIR3
D556             676  OPNSK3
D556 30 09       677   JR NC,OPNSK1
D558 3A 8C E0    678    LD A,(#STOP)
D55B FE 03       679    CP 3
D55D 28 09       680    JR Z,OPNSK4
D55F 37          681     SCF
D560 C9          682     RET
D561             683  OPNSK1 ;WRITE
D561 3A 89 E0    684   LD A,(#FDOP+2)
D564 B7          685   OR A
D565 C2 42 E0    686   JP NZ,#ERR7
D568             687  OPNSK4
D568 AF          688   XOR A
D569 32 8C E0    689   LD (#STOP),A
D56C 21 01 00    690   LD HL,1
D56F 22 95 E0    691   LD (#CRS),HL
D572 CD 21 E0    692   CALL #SEC00 ;空をさがす
D575 2A 95 E0    693   LD HL,(#CRS)
D578 22 7C E0    694   LD (#ECRS),HL
D57B B7          695   OR A
D57C C9          696   RET
D57D             697
D57D             698  ;/ データ入出力先決定 /
D57D             699  ;
D57D             700  ICPO
D57D 3A 9C E0    701   LD A,(#IOFG)
D580 3D          702   DEC A ;1..入力要求
D581 28 2D       703   JR Z,ICPENT2
D583 3D          704   DEC A ;2..出力要求
D584 28 13       705   JR Z,ICPENT1
D586             706  ICPOLP ;0..初期化
D586 3E 01       707   LD A,1
D588 32 80 D1    708   LD (#OD),A
D58B 2A A4 E0    709   LD HL,(#BADR)
D58E 22 91 E0    710   LD (#BF),HL ;ADR
D591 FD 21 98 D5 711   LD IY,ICPO2 ;
D595 C3 BB D5    712   JP DEVDRV   ;
D598             713  ICPO2  ;=CALL DEVDRV
D598 D8          714   RET C ;
D599             715  ICPENT1
D599 CD 09 D0    716   CALL #MSX1 ;変換
D59C 3A 9C E0    717   LD A,(#IOFG)
D59F FE 02       718   CP 2
D5A1 28 E3       719   JR Z,ICPOLP
D5A3 3E 02       720   LD A,2
D5A5 32 80 D1    721   LD (#OD),A
D5A8 FD 21 AF D5 722   LD IY,ICPO3 ;
D5AC C3 BB D5    723   JP DEVDRV   ;
D5AF             724  ICPO3  ;=CALL DEVDRV
D5AF D8          725   RET C ;
D5B0             726  ICPENT2
D5B0 3A 9C E0    727   LD A,(#IOFG)
D5B3 B7          728   OR A
D5B4 20 E3       729   JR NZ,ICPENT1
D5B6 AF          730   XOR A
D5B7 32 80 D1    731   LD (#OD),A
D5BA C9          732   RET
D5BB             733
D5BB             734  DEVDRV ;実際のJUMP
D5BB 21 8A D1    735   LD HL,#DDRV
D5BE CD 6C D7    736   CALL ODST
D5C1 7E          737   LD A,(HL)
D5C2 87          738   ADD A,A
D5C3 4F          739   LD C,A
D5C4 87          740   ADD A,A
D5C5 81          741   ADD A,C
D5C6 DD 21 40 EC 742   LD IX,SRDR
D5CA 4F          743   LD C,A
D5CB 06 00       744   LD B,0
D5CD DD 09       745   ADD IX,BC
D5CF DD 7E 03    746   LD A,(IX+3)
D5D2 B7          747   OR A
D5D3 C8          748   RET Z ;0..BASIC
D5D4 DD 6E 04    749   LD L,(IX+4)
D5D7 DD 66 05    750   LD H,(IX+5)
D5DA FD E5       751   PUSH IY ;
D5DC E9          752   JP (HL) ;CALL (HL)
D5DD             753
D5DD             754  DEVOI ;IN BF DISK専門
D5DD 3A 80 D1    755   LD A,(#OD)
D5E0 3D          756   DEC A
D5E1 32 A1 E0    757   LD (#FRWF),A ;R&W FLUG
D5E4 CD 76 D7    758   CALL ODST2
D5E7 E5          759   PUSH HL
D5E8 DD E1       760   POP IX
D5EA DD 6E 23    761   LD L,(IX+35) ;以下
D5ED DD 66 24    762   LD H,(IX+36) ;情報設定
D5F0 22 8D E0    763   LD (#FF),HL
D5F3 DD 6E 29    764   LD L,(IX+41)
D5F6 DD 66 2A    765   LD H,(IX+42)
D5F9 22 95 E0    766   LD (#CRS),HL
D5FC DD 7E 28    767   LD A,(IX+40)
D5FF 32 97 E0    768   LD (#MAC),A
D602 DD 7E 27    769   LD A,(IX+39)
D605 32 8B E0    770   LD (#DN),A
D608 DD E5       771   PUSH IX
D60A CD 1B E0    772   CALL #VAR
D60D CD 09 E0    773   CALL #CRSRW ;R&W
D610 DD E1       774   POP IX
D612 D8          775   RET C
D613 2A 95 E0    776   LD HL,(#CRS)
D616 DD 75 29    777   LD (IX+41),L
D619 DD 74 2A    778   LD (IX+42),H
D61C             779  NULRET
D61C AF          780   XOR A
D61D 32 A1 E0    781   LD (#FRWF),A
D620 C9          782   RET
D621             783
D621             784  ;(FILE-NAME展開)
D621             785  ;
D621             786  MOTHR ;IN (DE)
D621 CD BE D6    787   CALL CLRFAM ;MEM CLEAR
D624 78          788   LD A,B
D625 B7          789   OR A
D626 20 19       790   JR NZ,MOTHSK2 ;IF 長さ>0
D628 3A 81 D1    791    LD A,(#OP) ;1..'*.*'
D62B B7          792    OR A
D62C CA 36 E0    793    JP Z,#ERR3
D62F FE 02       794    CP 2        ;2..'.*'
D631 CA 36 E0    795    JP Z,#ERR3
D634 FE 03       796    CP 3
D636 C8          797    RET Z
D637 3E 2A       798     LD A,'*'
D639 32 C7 DD    799     LD (BGAM),A
D63C 32 D4 DD    800     LD (BGAM+13),A
D63F 18 03       801     JR MOTHSK3
D641             802  MOTHSK2
D641 CD 7B D6    803   CALL INSTEN
D644             804  MOTHSK3
D644 01 10 00    805   LD BC,16
D647 21 C7 DD    806   LD HL,BGAM
D64A 3A 81 D1    807   LD A,(#OP)
D64D FE 03       808   CP 3
D64F 28 07       809   JR Z,MOTHSK4
D651 11 54 E0    810    LD DE,#FLNM
D654 ED B0       811    LDIR
D656 B7          812    OR A
D657 C9          813    RET
D658             814
D658             815  MOTHSK4
D658 06 0D       816   LD B,13
D65A 11 68 E0    817   LD DE,#FNAM
D65D CD 6D D6    818   CALL SUGO2
D660 06 03       819   LD B,3
D662 21 D4 DD    820   LD HL,BGAM+13
D665 11 75 E0    821   LD DE,#FNAM+13
D668 CD 6D D6    822   CALL SUGO2
D66B B7          823   OR A
D66C C9          824   RET
D66D             825  SUGO2
D66D 7E          826   LD A,(HL)
D66E FE 2A       827   CP '*'
D670 C8          828   RET Z
D671 FE 3F       829   CP '?'
D673 28 01       830   JR Z,SUGOS
D675 12          831    LD (DE),A
D676             832  SUGOS
D676 23          833   INC HL
D677 13          834   INC DE
D678 10 F3       835   DJNZ SUGO2
D67A C9          836   RET
D67B             837
D67B             838  INSTEN ;以下 NAME 処理
D67B D5          839   PUSH DE
D67C 48          840   LD C,B ;C=DATA LENGTH
D67D 06 0D       841   LD B,13
D67F 21 C7 DD    842   LD HL,BGAM
D682             843  INSTENLP
D682 1A          844   LD A,(DE)
D683 13          845   INC DE
D684 FE 2E       846   CP '.'
D686 28 19       847   JR Z,INSTENSK
D688 77          848   LD (HL),A ;DATA SET
D689 23          849   INC HL
D68A 0D          850    DEC C
D68B 28 07       851    JR Z,INSTENS2
D68D 10 F3       852    DJNZ INSTENLP
D68F CD B4 D6    853   CALL INSTEN2 ;FIND '.'
D692 30 0D       854   JR NC,INSTENSK ;IF あれば JR
D694             855  INSTENS2 ;'.'がない
D694 D1          856   POP DE
D695 3A 81 D1    857   LD A,(#OP) ;
D698 FE 02       858   CP 2       ;(#OP)=2.拡張子='*'
D69A C0          859   RET NZ     ;
D69B 3E 2A       860   LD A,'*'   ;
D69D 32 D4 DD    861   LD (BGAM+13),A
D6A0 C9          862   RET
D6A1             863
D6A1             864  INSTENSK ;拡長子処理
D6A1 0D          865   DEC C
D6A2 28 0E       866   JR Z,INSTSS
D6A4 06 03       867   LD B,3
D6A6 21 D4 DD    868   LD HL,BGAM+13
D6A9             869  INSTLP2
D6A9 1A          870    LD A,(DE)
D6AA 77          871    LD (HL),A
D6AB 13          872    INC DE
D6AC 23          873    INC HL
D6AD 0D          874    DEC C
D6AE 28 02       875    JR Z,INSTSS
D6B0 10 F7       876    DJNZ INSTLP2
D6B2             877  INSTSS
D6B2 D1          878   POP DE
D6B3 C9          879   RET
D6B4             880
D6B4             881  INSTEN2 ;OVER 13文字
D6B4 1A          882   LD A,(DE)
D6B5 13          883   INC DE
D6B6 FE 2E       884   CP '.'
D6B8 C8          885   RET Z
D6B9 0D          886   DEC C
D6BA 20 F8       887   JR NZ,INSTEN2
D6BC 37          888   SCF
D6BD C9          889   RET
D6BE             890
D6BE             891  CLRFAM  ;WORK
D6BE C5          892   PUSH BC ;初期化
D6BF D5          893   PUSH DE
D6C0 E5          894   PUSH HL
D6C1 01 0F 00    895   LD BC,15
D6C4 3E 20       896   LD A,' '
D6C6 11 C8 DD    897   LD DE,BGAM+1
D6C9 21 C7 DD    898   LD HL,BGAM
D6CC 77          899   LD (HL),A
D6CD ED B0       900   LDIR
D6CF E1          901   POP HL
D6D0 D1          902   POP DE
D6D1 C1          903   POP BC
D6D2 C9          904   RET
D6D3             905
D6D3             906  ;/ CLOSE 処理 /
D6D3             907  ;
D6D3             908  CLOSE ;IN FNAM,FSZL
D6D3 CD 76 D7    909   CALL ODST2
D6D6 E5          910   PUSH HL ;
D6D7 DD E1       911   POP IX  ;LD IX,HL
D6D9 2A 86 D1    912   LD HL,(#FSZL) ;以下
D6DC DD 75 10    913   LD (IX+16),L  ;情報SET
D6DF DD 74 11    914   LD (IX+17),H
D6E2 2A 88 D1    915   LD HL,(#FSZH)
D6E5 DD 75 12    916   LD (IX+18),L
D6E8 DD 74 13    917   LD (IX+19),H
D6EB 2A A4 E0    918   LD HL,(#BADR)
D6EE 22 52 E0    919   LD (#MORF),HL
D6F1 DD 7E 2B    920   LD A,(IX+43)
D6F4 32 51 E0    921   LD (#DIRN),A
D6F7 DD 6E 2C    922   LD L,(IX+44)
D6FA DD 66 2D    923   LD H,(IX+45)
D6FD 22 95 E0    924   LD (#CRS),HL
D700 CD 1B D0    925   CALL #DIR3 ;NEXT NAME FIND
D703 2A A4 E0    926   LD HL,(#BADR)
D706 D4 36 D7    927   CALL NC,CLOSESK
D709 22 52 E0    928   LD (#MORF),HL
D70C AF          929   XOR A
D70D 32 8C E0    930   LD (#STOP),A
D710 CD D4 D9    931   CALL DSEC0 ;空DIRをさがす
D713 DA 3C E0    932   JP C,#ERR5
D716 CD 76 D7    933    CALL ODST2
D719 11 68 E0    934    LD DE,#FNAM
D71C 01 23 00    935    LD BC,35
D71F ED B0       936    LDIR
D721 21 80 D0    937    LD HL,#ZOKU1
D724 CD 6C D7    938    CALL ODST
D727 7E          939    LD A,(HL)
D728 32 A3 E0    940    LD (#ZOKU),A ;属性
D72B 3E 01       941   LD A,1        ;
D72D 32 A1 E0    942   LD (#FRWF),A ;WRITE
D730 CD 07 D9    943   CALL NAGRM    ;NAME SET
D733 C3 3C D7    944   JP CLOSE2
D736             945
D736             946  CLOSESK
D736 CD 03 E0    947   CALL #DLFAT   ;DEL FAT
D739 C3 12 D0    948   JP #DLDIR     ;DEL NAME
D73C             949
D73C             950  ;/ DISK CLOSE /
D73C             951  ;
D73C             952  CLOSE2
D73C CD 76 D7    953   CALL ODST2
D73F E5          954   PUSH HL ;
D740 DD E1       955   POP IX  ;LD IX,HL
D742 DD 7E 2B    956   LD A,(IX+43)
D745 32 51 E0    957   LD (#DIRN),A
D748 B7          958   OR A
D749 28 11       959   JR Z,CLOSE2SK ;ROOT
D74B DD 6E 2C    960   LD L,(IX+44)
D74E DD 66 2D    961   LD H,(IX+45)
D751 22 7C E0    962   LD (#ECRS),HL
D754 E5          963   PUSH HL ;DIRがはみだす
D755 CD 03 E0    964    CALL #DLFAT ;DEL FAT
D758 E1          965   POP HL  ;
D759 22 95 E0    966   LD (#CRS),HL
D75C             967  CLOSE2SK
D75C DD 6E 23    968   LD L,(IX+35)
D75F DD 66 24    969   LD H,(IX+36)
D762 22 8D E0    970   LD (#FF),HL
```

▶5月12日，ダンジョンマスターをやっていると，雨が降り始めた。そのうち雨にまじって白いものも降ってきた。霰かなと思っていると，つぶが大きくなってきたので雹だとわかった。おかげで，学校のガラスが500枚以上割れ14日が休みに。ピンポン玉ぐらいのものもあったそうで，まさに，天災は忘れた頃にやってきた。　佐藤　史彦（17）北海道

```
D765 CD 03 D0     971   CALL #DIRWT ;WRITE
D768 D4 06 E0     972   CALL NC,#FATWT ;
D76B C9           973   RET
D76C              974
D76C              975  ;/ HL=HL+(#OD) /
D76C              976  ;
D76C              977  ODST ;IN HL OUT HL
D76C C5           978   PUSH BC
D76D 3A 80 D1     979   LD A,(#OD)
D770 4F           980   LD C,A
D771 06 00        981   LD B,0
D773 09           982   ADD HL,BC
D774 C1           983   POP BC
D775 C9           984   RET
D776              985
D776              986  ;/ HL=HL+46*(#OD) /
D776              987  ;   46..FNAM1の大きさ
D776              988  ODST2
D776 3A 80 D1     989   LD A,(#OD)
D779 6F           990   LD L,A
D77A 26 00        991   LD H,0
D77C 11 2E 00     992   LD DE,46
D77F CD 27 E0     993   CALL #MLTHD
D782 11 84 D0     994   LD DE,#FNAM1
D785 19           995   ADD HL,DE
D786 C9           996   RET
D787              997
D787              998  ;/ READ DIR /
D787              999  ;
D787             1000  DIR ;IN FLNM
D787 21 54 E0    1001   LD HL,#FLNM
D78A 18 03       1002   JR DIR3SK
D78C             1003  DIR3
D78C 21 68 E0    1004   LD HL,#FNAM  ;#FLNM保存時
D78F             1005  DIR3SK
D78F 22 EA DD    1006   LD (BFNM),HL
D792 CD B7 D7    1007   CALL EXTND    ;'*'->'?'
D795 AF          1008   XOR A        ;
D796 32 A1 E0    1009   LD (#FRWF),A ;READ
D799 CD DB D7    1010   CALL DIRRW    ;
D79C 2A 91 E0    1011   LD HL,(#BF)
D79F 22 20 DE    1012   LD (SHIME),HL
D7A2 C3 2F D8    1013   JP DIR2        ;FIND NAME
D7A5             1014
D7A5             1015  ;/ WRITE DIR /
D7A5             1016  ;
D7A5             1017  DIRWT
D7A5 3E 01       1018   LD A,1         ;WRITE
D7A7 32 A1 E0    1019   LD (#FRWF),A ;
D7AA CD DB D7    1020   CALL DIRRW    ;
D7AD 3E 20       1021   LD A,32
D7AF 32 98 E0    1022   LD (#PASS),A ;PASSWORD
D7B2 AF          1023   XOR A
D7B3 32 A1 E0    1024   LD (#FRWF),A
D7B6 C9          1025   RET
D7B7             1026
D7B7             1027  ;/ "*"を"?"に展開 /
D7B7             1028  ;
D7B7             1029  EXTND ;IN FLNM OUT FLNM
D7B7 C5          1030   PUSH BC
D7B8 D5          1031   PUSH DE
D7B9 E5          1032   PUSH HL
D7BA 2A EA DD    1033   LD HL,(BFNM)
D7BD 0E 02       1034   LD C,2
D7BF 06 0D       1035   LD B,13
D7C1             1036  EXTNDLP1
D7C1 7E          1037    LD A,(HL)
D7C2 FE 2A       1038    CP '*'
D7C4 28 05       1039    JR Z,EXTNDSK1
D7C6 23          1040    INC HL
D7C7 10 F8       1041    DJNZ EXTNDLP1
D7C9 18 06       1042   JR EXTNDSK2
D7CB             1043  EXTNDSK1
D7CB 3E 3F       1044   LD A,'?'
D7CD 77          1045   LD (HL),A
D7CE 23          1046   INC HL
D7CF 10 FA       1047   DJNZ EXTNDSK1
D7D1             1048  EXTNDSK2 ;拡長子
D7D1 06 03       1049   LD B,3
D7D3 0D          1050   DEC C
D7D4 20 EB       1051   JR NZ,EXTNDLP1
D7D6 B7          1052   OR A
D7D7 E1          1053   POP HL
D7D8 D1          1054   POP DE
D7D9 C1          1055   POP BC
D7DA C9          1056   RET
D7DB             1057
D7DB             1058  ;(READ & WRITE DIR)
D7DB             1059  ;
D7DB             1060  DIRRW ;IN BSIZ,DIRN,CRS
D7DB 2A A4 E0    1061   LD HL,(#BADR)
D7DE 22 91 E0    1062   LD (#BF),HL
D7E1 3A 51 E0    1063   LD A,(#DIRN)
D7E4 B7          1064   OR A
D7E5 C2 09 E0    1065   JP NZ,#CRSRW ;階層時
D7E8 C5          1066   PUSH BC
D7E9 CD F8 D7    1067   CALL RDIRD    ;FORMAT別設定
D7EC             1068  RTDIRLP
D7EC CD 00 E0    1069    CALL #RWREC ;R&W 1レコード
D7EF 38 05       1070    JR C,RDLPEXT
D7F1 0D          1071    DEC C         ;DIRレコード長
D7F2 28 02       1072    JR Z,RDLPEXT
D7F4 10 F6       1073    DJNZ RTDIRLP
D7F6             1074  RDLPEXT
D7F6 C1          1075   POP BC
D7F7 C9          1076   RET
D7F8             1077
D7F8             1078  RDIRD
D7F8 D5          1079   PUSH DE
D7F9 E5          1080   PUSH HL
D7FA 3A 97 E0    1081   LD A,(#MAC) ;FORMAT
D7FD DD 21 34 E1 1082   LD IX,RDDT1
D801 3D          1083   DEC A ;1..X12HD
D802 28 12       1084   JR Z,RDIRDSK
D804 DD 21 36 E1 1085   LD IX,RDDT2
D808 3D          1086   DEC A ;2..MS-DOS2HD
D809 28 0B       1087   JR Z,RDIRDSK
D80B DD 21 38 E1 1088   LD IX,RDDT3
D80F 3D          1089   DEC A ;3..X12D
D810 28 04       1090   JR Z,RDIRDSK
D812 DD 21 3A E1 1091   LD IX,RDDT4 ;4..MS-DOOS2D
D816             1092  RDIRDSK
D816 DD 6E 00    1093   LD L,(IX+0)
D819 26 00       1094   LD H,0
D81B 22 93 E0    1095   LD (#REC),HL ;レコード
D81E 2A 8F E0    1096   LD HL,(#BSIZ)
D821 ED 5B 9F E0 1097   LD DE,(#SLNG)
D825 CD 2A E0    1098   CALL #DIVHD
D828 47          1099   LD B,A
D829 DD 4E 01    1100   LD C,(IX+1) ;レコード長
D82C E1          1101   POP HL
D82D D1          1102   POP DE
D82E C9          1103   RET
D82F             1104
D82F             1105  ;(概当するFILE-NAMEをさがす)
D82F             1106  ;
D82F             1107  DIR2 ;IN MORF OUT MORF,DTBL
D82F CD 38 D8    1108   CALL SFLNM
D832 CA 07 D9    1109   JP Z,NAGRM ;NAME SET
D835 C3 36 E0    1110   JP #ERR3    ;一致なし
D838             1111
D838             1112  ;(FILE-NAMEが一致するか?)
D838             1113  ;
D838             1114  SFLNM ;IN MORF OUT MORF
D838 DD E5       1115   PUSH IX
D83A CD 53 DD    1116   CALL SETFNDT ;FORMAT別設定
D83D 06 80       1117   LD B,128 ;ファイルは128まで
D83F 2A 52 E0    1118   LD HL,(#MORF)
D842             1119  SFLNMLP
D842 ED 5B 20 DE 1120    LD DE,(SHIME) ;DIRの最後
```

```
D846 B7          1121    OR A  ;
D847 ED 52       1122    SBC HL,DE ;
D849 30 28       1123    JR NC,SFLED ;
D84B 19          1124    ADD HL,DE ;=LD HL,(#MORF)
D84C CD 12 E0    1125    CALL #LDAHL
D84F DD BE 00    1126    CP (IX+0) ;END
D852 28 1F       1127    JR Z,SFLED
D854 DD BE 01    1128    CP (IX+1) ;KILL
D857 28 14       1129    JR Z,NMKSK
D859 C5          1130     PUSH BC
D85A E5          1131     PUSH HL
D85B CD A5 D8    1132     CALL DZGET  ;属性
D85E CD B0 D8    1133     CALL SKDIR  ;SKIP?
D861 CC C2 D8    1134     CALL Z,SKDIR2 ;SKIP?
D864 E1          1135     POP HL
D865 E5          1136     PUSH HL
D866 CC 7E D8    1137     CALL Z,NIDFY  ;NAME?
D869 E1          1138     POP HL
D86A C1          1139     POP BC
D86B 28 0B       1140     JR Z,SFLNMSK  ;発見
D86D             1141  NMKSK
D86D 11 20 00    1142    LD DE,32      ;NEXT
D870 19          1143    ADD HL,DE     ;
D871 10 CF       1144    DJNZ SFLNMLP ;
D873             1145  SFLED
D873 AF          1146   XOR A
D874 3C          1147   INC A ;RZF
D875 2A 52 E0    1148   LD HL,(#MORF)
D878             1149  SFLNMSK
D878 22 52 E0    1150   LD (#MORF),HL
D87B DD E1       1151   POP IX
D87D C9          1152   RET
D87E             1153
D87E             1154  NIDFY ;IN HL OUT Z FLAG
D87E DD 5E 02    1155   LD E,(IX+2)
D881 16 00       1156   LD D,0
D883 19          1157   ADD HL,DE
D884 CD 12 E0    1158   CALL #LDAHL
D887 D6 2E       1159   SUB '.'
D889 20 02       1160   JR NZ,NIDSSK
D88B 3C          1161    INC A
D88C C9          1162    RET
D88D             1163  NIDSSK
D88D ED 5B EA DD 1164   LD DE,(BFNM)
D891 EB          1165   EX DE,HL ;HL=#FLNM,DE=BUFF
D892 DD 46 03    1166   LD B,(IX+3) ;B=LENGTH
D895 CD E1 D8    1167   CALL NIDLP  ;NAME?
D898 C0          1168   RET NZ
D899 2A EA DD    1169   LD HL,(BFNM)
D89C 01 0D 00    1170   LD BC,13
D89F 09          1171   ADD HL,BC
D8A0 06 03       1172   LD B,3         ;B=LENGTH
D8A2 C3 E1 D8    1173   JP NIDLP       ;NAME?
D8A5             1174
D8A5             1175  DZGET
D8A5 DD 5E 04    1176   LD E,(IX+4)
D8A8 16 00       1177   LD D,0
D8AA 19          1178   ADD HL,DE
D8AB CD 12 E0    1179   CALL #LDAHL
D8AE 47          1180   LD B,A
D8AF C9          1181   RET
D8B0             1182
D8B0             1183  SKDIR
D8B0 3A 65 E0    1184   LD A,(#SBDR)
D8B3 B7          1185   OR A
D8B4 C8          1186   RET Z ;0
D8B5 DD 4E 07    1187   LD C,(IX+7)
D8B8 3D          1188   DEC A
D8B9 28 04       1189   JR Z,SKDIR11
D8BB 78          1190   LD A,B
D8BC A1          1191   AND C
D8BD A9          1192   XOR C
D8BE C9          1193   RET ;D2
D8BF             1194  SKDIR11
D8BF 78          1195   LD A,B
D8C0 A1          1196   AND C
D8C1 C9          1197   RET ;D1
D8C2             1198
D8C2             1199  SKDIR2
D8C2 3A 64 E0    1200   LD A,(#DPNF)
D8C5 B7          1201   OR A
D8C6 C8          1202   RET Z ;0
D8C7 FE 02       1203   CP 2
D8C9 28 09       1204   JR Z,SKDIR22
D8CB FE 03       1205   CP 3
D8CD 28 0A       1206   JR Z,SKDIR23
D8CF 78          1207   LD A,B
D8D0 DD A6 05    1208   AND (IX+5) ;S1
D8D3 C0          1209   RET NZ
D8D4             1210  SKDIR22
D8D4 78          1211   LD A,B
D8D5 DD A6 06    1212   AND (IX+6) ;V12
D8D8 C9          1213   RET
D8D9             1214  SKDIR23
D8D9 78          1215   LD A,B
D8DA DD A6 06    1216   AND (IX+6)
D8DD DD AE 06    1217   XOR (IX+6)
D8E0 C9          1218   RET
D8E1             1219
D8E1             1220  NIDLP ;概当NAMEか?
D8E1 7E          1221    LD A,(HL)
D8E2 FE 3F       1222    CP '?'
D8E4 28 0B       1223    JR Z,NIDWILD
D8E6 CD 15 E0    1224    CALL #LDADE
D8E9 BE          1225    CP (HL)
D8EA 28 05       1226    JR Z,NIDWILD
D8EC CD F7 D8    1227     CALL OKCHG ;大小変換
D8EF BE          1228     CP (HL)
D8F0 C0          1229     RET NZ ;NZ=違う
D8F1             1230  NIDWILD
D8F1 23          1231    INC HL
D8F2 13          1232    INC DE
D8F3 10 EC       1233    DJNZ NIDLP
D8F5 AF          1234    XOR A    ;Z=概当する
D8F6 C9          1235   RET
D8F7             1236
D8F7             1237  ;(大文字 & 小文字変換)
D8F7             1238  ; A->a    z->Z
D8F7             1239  OKCHG ;IN A OUT A
D8F7 E5          1240   PUSH HL
D8F8 67          1241   LD H,A
D8F9 E6 20       1242   AND $20
D8FB 20 05       1243   JR NZ,OKCHGSK
D8FD 3E 20       1244    LD A,$20 ;&B00100000
D8FF B4          1245    OR H ;大->小
D900 18 03       1246    JR OKCHGRT
D902             1247  OKCHGSK
D902 3E DF       1248    LD A,$DF ;&B11011111
D904 A4          1249    AND H ;小->大
D905             1250  OKCHGRT
D905 E1          1251   POP HL
D906 C9          1252   RET
D907             1253
D907             1254  ;(FILE情報をWORKにSET)
D907             1255  ;
D907             1256  NAGRM ;IN MORF OUT DTBL
D907 CD 96 D9    1257   CALL FMSETS ;CLEAR
D90A             1258
D90A 3A 97 E0    1259   LD A,(#MAC) ;FORMAT別
D90D 21 4C E1    1260   LD HL,NADT1
D910 3D          1261   DEC A
D911 28 0F       1262   JR Z,NASKMC
D913 21 72 E1    1263   LD HL,NADT2
D916 3D          1264   DEC A
D917 28 09       1265   JR Z,NASKMC
D919 21 4C E1    1266   LD HL,NADT1
D91C 3D          1267   DEC A
D91D 28 03       1268   JR Z,NASKMC
D91F 21 72 E1    1269   LD HL,NADT2
D922             1270  NASKMC
```

```
D922             1271
D922 46          1272   LD B,(HL) ;SET DATA数
D923 23          1273   INC HL
D924             1274  NAGRLP2
D924 C5          1275    PUSH BC
D925 4E          1276    LD C,(HL) ;BC=位置
D926 06 00       1277    LD B,0     ;
D928 23          1278    INC HL
D929 7E          1279    LD A,(HL) ;A=長さ
D92A 23          1280    INC HL
D92B 5E          1281    LD E,(HL) ;DE=
D92C 23          1282    INC HL     ;SETするADR
D92D 56          1283    LD D,(HL) ;
D92E 23          1284    INC HL
D92F E5          1285    PUSH HL
D930 2A 52 E0    1286     LD HL,(#MORF)
D933 09          1287     ADD HL,BC  ;(HL)から
D934 47          1288     LD B,A     ;(DE)へ
D935 CD B5 D9    1289     CALL JITTS ;B回 SET
D938 E1          1290    POP HL
D939 C1          1291    POP BC
D93A 10 E8       1292    DJNZ NAGRLP2
D93C CD 4B D9    1293   CALL NADIN ;属性
D93F 2A 52 E0    1294   LD HL,(#MORF)
D942 11 20 00    1295   LD DE,32  ;
D945 19          1296   ADD HL,DE ;NEXT
D946 22 52 E0    1297   LD (#MORF),HL
D949 B7          1298   OR A
D94A C9          1299   RET
D94B             1300
D94B             1301  ;(FILE属性をWORKにSET)
D94B             1302  ;
D94B             1303  NADIN
D94B 4E          1304   LD C,(HL)
D94C 23          1305   INC HL
D94D 06 00       1306   LD B,0
D94F EB          1307   EX DE,HL
D950 2A 52 E0    1308   LD HL,(#MORF)
D953 09          1309   ADD HL,BC
D954 3A A1 E0    1310    LD A,(#FRWF) ;
D957 B7          1311    OR A          ;
D958 20 15       1312    JR NZ,NADIN2 ;WRITE
D95A CD 12 E0    1313   CALL #LDAHL
D95D 4F          1314   LD C,A        ;BIT位置
D95E 21 87 E0    1315   LD HL,#FDOP ;WORK
D961 06 04       1316   LD B,4        ;4バイト
D963             1317  NADINLP
D963 1A          1318    LD A,(DE)
D964 13          1319    INC DE
D965 A1          1320    AND C ;BIT取出
D966 77          1321    LD (HL),A
D967 23          1322    INC HL
D968 10 F9       1323    DJNZ NADINLP
D96A 79          1324   LD A,C ;属性そのもの
D96B 32 A3 E0    1325   LD (#ZOKU),A
D96E C9          1326   RET
D96F             1327
D96F             1328  NADIN2 ;WRITE
D96F E5          1329   PUSH HL ;MORF+
D970 21 87 E0    1330   LD HL,#FDOP
D973 EB          1331   EX DE,HL
D974 06 04       1332   LD B,4
D976 0E 00       1333   LD C,0
D978             1334  NADINLP2
D978 1A          1335    LD A,(DE) ;FDOP
D979 B7          1336    OR A
D97A 28 03       1337    JR Z,NADINSK2
D97C 79          1338     LD A,C  ;
D97D B6          1339     OR (HL) ;C=C OR (HL)
D97E 4F          1340     LD C,A  ;
D97F             1341  NADINSK2
D97F 13          1342    INC DE
D980 23          1343    INC HL
D981 10 F5       1344    DJNZ NADINLP2
D983 D1          1345   POP DE ;MORF+
D984 CD 8B D9    1346   CALL ZOKUS
D987 79          1347   LD A,C
D988 C3 18 E0    1348   JP #LDDEA
D98B             1349
D98B             1350  ZOKUS
D98B 3A A3 E0    1351   LD A,(#ZOKU)
D98E B7          1352   OR A
D98F C8          1353   RET Z
D990 4F          1354   LD C,A
D991 AF          1355   XOR A
D992 32 A3 E0    1356   LD (#ZOKU),A
D995 C9          1357   RET
D996             1358
D996             1359  ;(エリアをCLEAR)
D996             1360  ;
D996             1361  FMSETS
D996 3A A1 E0    1362    LD A,(#FRWF)
D999 B7          1363    OR A
D99A 20 0B       1364    JR NZ,FMSETS2 ;WRITE
D99C 21 68 E0    1365   LD HL,#FNAM
D99F 06 23       1366   LD B,35
D9A1 AF          1367   XOR A
D9A2             1368  DTBLCL
D9A2 77          1369    LD (HL),A
D9A3 23          1370    INC HL
D9A4 10 FC       1371    DJNZ DTBLCL
D9A6 C9          1372   RET
D9A7             1373
D9A7             1374  FMSETS2 ;WRITE
D9A7 ED 5B 52 E0 1375   LD DE,(#MORF)
D9AB 06 20       1376   LD B,32
D9AD             1377  RVFMCL
D9AD AF          1378    XOR A
D9AE CD 18 E0    1379    CALL #LDDEA
D9B1 13          1380    INC DE
D9B2 10 F9       1381    DJNZ RVFMCL
D9B4 C9          1382   RET
D9B5             1383
D9B5             1384  ;(Bバイト分をエリアにSET)
D9B5             1385  ;
D9B5             1386  JITTS
D9B5 3A A1 E0    1387   LD A,(#FRWF)
D9B8 B7          1388   OR A
D9B9 20 0C       1389   JR NZ,JITTS2 ;WRITE
D9BB             1390  JITTSLP
D9BB 79          1391   LD A,C
D9BC FE 20       1392   CP 32
D9BE C4 12 E0    1393   CALL NZ,#LDAHL
D9C1 12          1394   LD (DE),A
D9C2 23          1395   INC HL
D9C3 13          1396   INC DE
D9C4 10 F5       1397   DJNZ JITTSLP
D9C6 C9          1398   RET
D9C7             1399
D9C7             1400  JITTS2 ;WRITE
D9C7 EB          1401   EX DE,HL
D9C8             1402  NAGRLP
D9C8 79          1403    LD A,C
D9C9 FE 20       1404    CP 32
D9CB 7E          1405    LD A,(HL)
D9CC C4 18 E0    1406    CALL NZ,#LDDEA
D9CF 23          1407    INC HL
D9D0 13          1408    INC DE
D9D1 10 F5       1409    DJNZ NAGRLP
D9D3 C9          1410   RET
D9D4             1411
D9D4             1412  ;(空DIRをさがす)
D9D4             1413  ;
D9D4             1414  DSEC0 ;IN MORF OUT MORF
D9D4 CD 53 DD    1415   CALL SETFNDT ;FORMAT別
D9D7 06 00       1416   LD B,0
D9D9 2A 52 E0    1417   LD HL,(#MORF)
D9DC 11 20 00    1418   LD DE,32
D9DF             1419  DSECLP
D9DF CD 12 E0    1420    CALL #LDAHL
```

▶PC-6001ユーザーだった頃，ナイコンの友人 Tel にばかにされ「ナイコンのぶんざいで」と思っていたらX68000を買った。だが，本日 ACE-HDを買い，大声で言う。「HDをつけてみろ！」これで私もX68000のユーザーだ。　　城之内　一茂（18）千葉県

```
D9E2 DD BE 00      1421    CP (IX+0) ;END
D9E5 28 0B         1422    JR Z,DSECRT2
D9E7 DD BE 01      1423    CP (IX+1) ;KILL
D9EA 28 1C         1424    JR Z,DSECRT
D9EC 19            1425    ADD HL,DE    ;NEXT
D9ED 10 F0         1426    DJNZ DSECLP ;
D9EF C3 3C E0      1427   JP #ERR5
D9F2               1428  DSECRT2
D9F2 E5            1429   PUSH HL
D9F3 19            1430   ADD HL,DE
D9F4 DD 7E 00      1431   LD A,(IX+0)
D9F7 EB            1432   EX DE,HL     ;END
D9F8 CD 18 E0      1433   CALL #LDDEA ;CODE
D9FB EB            1434   EX DE,HL     ;WRITE
D9FC 23            1435   INC HL
D9FD ED 5B A4 E0   1436   LD DE,(#BADR) ;ADRの
DA01 B7            1437   OR A          ;つじつま
DA02 ED 52         1438   SBC HL,DE     ;合わせ
DA04 22 9D E0      1439   LD (#MSBT),HL ;
DA07 E1            1440   POP HL
DA08               1441  DSECRT ;みつかる
DA08 22 52 E0      1442   LD (#MORF),HL ;位置
DA0B C9            1443   RET
DA0C               1444
DA0C               1445  ;/ DIRを消去 /
DA0C               1446  ;
DA0C               1447  DELDIR ;IN MORF
DA0C CD 53 DD      1448   CALL SETFNDT ;FORMAT別
DA0F 2A 52 E0      1449   LD HL,(#MORF)
DA12 11 20 00      1450   LD DE,32  ;
DA15 B7            1451   OR A      ;
DA16 ED 52         1452   SBC HL,DE ;1つ前へ
DA18 54            1453   LD D,H    ;もどす
DA19 5D            1454   LD E,L    ;
DA1A DD 7E 01      1455   LD A,(IX+1)
DA1D C3 18 E0      1456   JP #LDDEA ;KILL
DA20               1457
DA20               1458  ;/ 空クラスターを調べる /
DA20               1459  ;
DA20               1460  DFRE ;OUT CRS
DA20 21 00 00      1461   LD HL,0
DA23 22 EC DD      1462   LD (FREE),HL
DA26 23            1463   INC HL
DA27 22 95 E0      1464   LD (#CRS),HL
DA2A               1465  FREELP
DA2A CD 21 E0      1466    CALL #SEC00 ;さがす
DA2D 38 09         1467    JR C,FREERT ;IF ない JR
DA2F 2A EC DD      1468    LD HL,(FREE)
DA32 23            1469    INC HL
DA33 22 EC DD      1470    LD (FREE),HL
DA36 18 F2         1471    JR FREELP
DA38               1472  FREERT
DA38 AF            1473   XOR A
DA39 32 8C E0      1474   LD (#STOP),A
DA3C 2A EC DD      1475   LD HL,(FREE) ;空数を
DA3F 22 95 E0      1476   LD (#CRS),HL ;(#CRS)へ
DA42 C9            1477   RET
DA43               1478
DA43               1479  ;/ DATA 変換 /
DA43               1480  ;
DA43               1481  MSX1 ;IN BADR,BSIZ,MSBT, EDR/W
                         IOFG
DA43 3A E5 EC      1482    LD A,(&ESCP) ;
DA46 E6 0F         1483    AND $F       ;
DA48 CA 7A DB      1484    JP Z,NOCHECK ;変換しない
DA4B 3A 9C E0      1485   LD A,(#IOFG)
DA4E 3D            1486   DEC A
DA4F CA CC DA      1487   JP Z,OUTPUT2  ;OUPUT 要求
DA52 3D            1488   DEC A
DA53 CA FD DA      1489   JP Z,INPUT2   ;INPUT 要求
DA56 CD 3A DB      1490    CALL DLYSET  ;初期化
DA59               1491  MSXLP1
DA59 3A EE DD      1492    LD A,(NXTDB) ;
DA5C FE 0D         1493    CP $0D       ;
DA5E 28 3B         1494    JR Z,MAECR   ;前のデータ=$0D?
DA60               1495  MAERT
DA60 CD 15 E0      1496   CALL #LDADE
DA63 CD 58 DB      1497   CALL ENDCODE  ;現在のデータ
DA66 38 1C         1498   JR C,MSXSK1   ;=END-CODE?
DA68 FE 0D         1499   CP $0D        ;=$0D?
DA6A 20 10         1500   JR NZ,IMACR   ;
DA6C 32 EE DD      1501    LD (NXTDB),A ;$0Dなら保存
DA6F 3A E5 EC      1502    LD A,(&ESCP)
DA72 E6 0F         1503    AND $F
DA74 FE 03         1504    CP 3
DA76 3E 0D         1505    LD A,$0D
DA78 20 0A         1506    JR NZ,MSXSK1
DA7A 18 0A         1507    JR MSXRT1      ;POINTER+
DA7C               1508  IMACR
DA7C 21 8E D1      1509   LD HL,#CTRL    ;
DA7F BE            1510   CP (HL)        ;CONTROL-CODE
DA80 30 02         1511   JR NC,MSXSK1   ;
DA82 3E 20         1512    LD A,' '      ;$20以下=$20
DA84               1513  MSXSK1
DA84 13            1514   INC DE         ;入力用
DA85 0B            1515   DEC BC         ;+POINTER
DA86               1516  MSXRT1
DA86 D9            1517   EXX
DA87 CD 18 E0      1518   CALL #LDDEA    ;出力用
DA8A 13            1519   INC DE         ;+POINTER
DA8B 0B            1520   DEC BC         ;
DA8C D9            1521   EXX
DA8D               1522  MSXRT2
DA8D 78            1523   LD A,B         ;入力DATA
DA8E B1            1524   OR C           ;END?
DA8F CA E1 DA      1525   JP Z,INPUT     ;IF END
DA92               1526  MSXRT3
DA92 D9            1527   EXX            ;出力用BUFF
DA93 78            1528   LD A,B         ;は一杯か?
DA94 B1            1529   OR C           ;
DA95 D9            1530   EXX            ;
DA96 CA B7 DA      1531   JP Z,OUTPUT    ;IF 一杯
DA99               1532  MSXRT4
DA99 18 BE         1533   JR MSXLP1      ;LOOP
DA9B               1534
DA9B               1535  MAECR           ;前データ=$0D
DA9B AF            1536   XOR A
DA9C 32 EE DD      1537   LD (NXTDB),A
DA9F 3A E5 EC      1538    LD A,(&ESCP)
DAA2 E6 0F         1539    AND $F
DAA4 FE 03         1540    CP 3
DAA6 38 04         1541    JR C,CRLF
DAA8 3E 0A         1542     LD A,$0A     ;$0Aを挿入
DAAA 18 D8         1543     JR MSXSK1
DAAC               1544  CRLF            ;$0Aを抜く
DAAC CD 15 E0      1545   CALL #LDADE    ;今データ=$0A?
DAAF FE 0A         1546   CP $0A         ;
DAB1 20 AD         1547   JR NZ,MAERT    ;IF 違う
DAB3 13            1548   INC DE         ;
DAB4 0B            1549   DEC BC         ;+POINTER
DAB5 18 D6         1550   JR MSXRT2
DAB7               1551
DAB7               1552  ;
DAB7               1553  OUTPUT ;出力要求してRET
DAB7 ED 53 EF DD   1554   LD (BFA),DE ;保存
DABB ED 43 F1 DD   1555   LD (BSA),BC ;
DABF CD 22 DB      1556   CALL LDMSBT ;DATA SET
DAC2 AF            1557   XOR A
DAC3 32 99 E0      1558   LD (#EDW),A
DAC6 3E 01         1559   LD A,1        ;出力
DAC8 32 9C E0      1560   LD (#IOFG),A ;
DACB C9            1561   RET      ;ここから
DACC               1562  OUTPUT2 ;ここへ戻ってくる
DACC ED 5B EF DD   1563   LD DE,(BFA)  ;レジスター
DAD0 ED 4B F1 DD   1564   LD BC,(BSA)  ;復帰
DAD4 D9            1565   EXX
DAD5 2A A4 E0      1566   LD HL,(#BADR) ;
DAD8 ED 4B 8F E0   1567   LD BC,(#BSIZ) ;
DADC 09            1568   ADD HL,BC
DADD EB            1569   EX DE,HL
DADE D9            1570   EXX
DADF 18 B8         1571   JR MSXRT4
DAE1               1572
DAE1               1573  INPUT ;入力要求してRET
DAE1 D9            1574   EXX
DAE2 ED 53 EF DD   1575   LD (BFA),DE ;保存
DAE6 ED 43 F1 DD   1576   LD (BSA),BC ;
DAEA D9            1577   EXX
DAEB 2A A4 E0      1578   LD HL,(#BADR)
DAEE 22 91 E0      1579   LD (#BF),HL
DAF1 3A 9B E0      1580    LD A,(#EDR)  ;DATA-END?
DAF4 B7            1581    OR A         ;
DAF5 20 1E         1582    JR NZ,REALRT ;IF END
DAF7 3E 02         1583   LD A,2
DAF9 32 9C E0      1584   LD (#IOFG),A
DAFC C9            1585   RET
DAFD               1586  INPUT2
DAFD CD AD DB      1587   CALL SBCFZ ;FILE-SIZE
DB00 ED 4B 9D E0   1588   LD BC,(#MSBT) ;復帰
DB04 ED 5B A4 E0   1589   LD DE,(#BADR)
DB08 D9            1590   EXX
DB09 ED 5B EF DD   1591   LD DE,(BFA)
DB0D ED 4B F1 DD   1592   LD BC,(BSA)
DB11 D9            1593   EXX
DB12 C3 92 DA      1594   JP MSXRT3
DB15               1595
DB15               1596  REALRT ;終了
DB15 CD 22 DB      1597   CALL LDMSBT
DB18               1598  REALRT2
DB18 3E 01         1599   LD A,1
DB1A 32 99 E0      1600   LD (#EDW),A
DB1D AF            1601   XOR A ;これで終わりの意
DB1E 32 9C E0      1602   LD (#IOFG),A
DB21 C9            1603   RET
DB22               1604
DB22               1605  LDMSBT ;#BFと#MSBTにSET
DB22 D9            1606   EXX
DB23 2A A4 E0      1607   LD HL,(#BADR)
DB26 ED 4B 8F E0   1608   LD BC,(#BSIZ)
DB2A 09            1609   ADD HL,BC
DB2B 22 91 E0      1610   LD (#BF),HL ;DATAの先頭ADR
DB2E EB            1611   EX DE,HL
DB2F B7            1612   OR A
DB30 ED 52         1613   SBC HL,DE
DB32 22 9D E0      1614   LD (#MSBT),HL ;DATAのLENGTH
DB35 CD 11 DC      1615   CALL FLSZSET
DB38 D9            1616   EXX
DB39 C9            1617   RET
DB3A               1618
DB3A               1619  ;(WORK & レジスター初期化)
DB3A               1620  ;
DB3A               1621  DLYSET ;IN BADR,BSIZ,MSBT
DB3A CD E0 DB      1622   CALL FLSZCLR ;F-SIZE CLEAR
DB3D AF            1623   XOR A
DB3E 32 EE DD      1624   LD (NXTDB),A
DB41 CD AD DB      1625   CALL SBCFZ
DB44 ED 4B 9D E0   1626   LD BC,(#MSBT)
DB48 ED 5B A4 E0   1627   LD DE,(#BADR)
DB4C D9            1628   EXX
DB4D 2A A4 E0      1629   LD HL,(#BADR)
DB50 ED 4B 8F E0   1630   LD BC,(#BSIZ)
DB54 09            1631   ADD HL,BC
DB55 EB            1632   EX DE,HL
DB56 D9            1633   EXX
DB57 C9            1634   RET
DB58               1635
DB58               1636  ;(END-CODE 判定&変換)
DB58               1637  ;
DB58               1638  ENDCODE
DB58 21 E7 EC      1639   LD HL,&EOF2
DB5B BE            1640   CP (HL)      ;END?
DB5C 28 0C         1641   JR Z,ENDSCF ;IF END
DB5E 08            1642    EX AF,AF'
DB5F 3A E5 EC      1643    LD A,(&ESCP)
DB62 E6 0F         1644    AND $F
DB64 3D            1645    DEC A
DB65 28 10         1646    JR Z,ONLYEOF
DB67 08            1647    EX AF,AF'
DB68 B7            1648   OR A
DB69 C9            1649   RET
DB6A               1650  ENDSCF ;END-CODE
DB6A 3E 01         1651   LD A,1          ;終了指令
DB6C 32 9B E0      1652   LD (#EDR),A  ;
DB6F 3A E8 EC      1653   LD A,(&EOF3) ;変換
DB72 01 01 00      1654   LD BC,1
DB75 37            1655   SCF
DB76 C9            1656   RET
DB77               1657  ONLYEOF ;END-CODEのみ変換
DB77 08            1658   EX AF,AF'
DB78 37            1659   SCF
DB79 C9            1660   RET
DB7A               1661
DB7A               1662  ;/ DATA変換なしの時の処理 /
DB7A               1663  ;
DB7A               1664  NOCHECK
DB7A 2A A4 E0      1665   LD HL,(#BADR)
DB7D 22 91 E0      1666   LD (#BF),HL
DB80 3A 9C E0      1667   LD A,(#IOFG)
DB83 3D            1668   DEC A ;1..出力
DB84 28 21         1669   JR Z,NOCI
DB86 3D            1670   DEC A ;2..入力
DB87 28 0F         1671   JR Z,NOCO2
DB89 CD E0 DB      1672    CALL FLSZCLR
DB8C 18 0A         1673    JR NOCO2
DB8E               1674  NOCO
DB8E AF            1675   XOR A
DB8F 32 99 E0      1676   LD (#EDW),A
DB92 3E 01         1677   LD A,1
DB94 32 9C E0      1678   LD (#IOFG),A ;出力
DB97 C9            1679   RET
DB98               1680  NOCO2
DB98 CD AD DB      1681   CALL SBCFZ   ;SIZE CHECK
DB9B CD 11 DC      1682   CALL FLSZSET ;SIZE SET
DB9E 3A 9B E0      1683    LD A,(#EDR)    ;END?
DBA1 B7            1684    OR A           ;
DBA2 C2 18 DB      1685    JP NZ,REALRT2 ;IF END
DBA5 18 E7         1686   JR NOCO
DBA7               1687
DBA7               1688  NOCI     ;入力要求
DBA7 3E 02         1689   LD A,2 ;
DBA9 32 9C E0      1690   LD (#IOFG),A
DBAC C9            1691   RET
DBAD               1692
DBAD               1693  ;(FILE-SIZEによる終了判定)
DBAD               1694  ;
DBAD               1695  SBCFZ
DBAD 3A E5 EC      1696   LD A,(&ESCP)
DBB0 FE 10         1697   CP $10
DBB2 D8            1698   RET C ;IF SIZE指定がない
DBB3 2A F3 DD      1699   LD HL,(SUBSL) ;下位16BIT
DBB6 44            1700   LD B,H
DBB7 4D            1701   LD C,L
DBB8 ED 5B 8F E0   1702   LD DE,(#BSIZ) ;BUFF-SIZE
DBBC B7            1703   OR A
DBBD ED 52         1704   SBC HL,DE
DBBF 22 F3 DD      1705   LD (SUBSL),HL
DBC2 2A F5 DD      1706   LD HL,(SUBSH) ;上位16BIT
DBC5 11 00 00      1707   LD DE,0
DBC8 28 11         1708   JR Z,ZBASAZS ;IF SIZE=BUFF
DBCA ED 52         1709   SBC HL,DE
DBCC 22 F5 DD      1710   LD (SUBSH),HL
DBCF D0            1711   RET NC ;IF まだ残っている
DBD0               1712  ZBSAZ   ;F-SIZE終わり
DBD0 3A 9B E0      1713   LD A,(#EDR) ;
DBD3 B7            1714   OR A        ;
DBD4 28 32         1715   JR Z,ZBSAZ2 ;FATはまだある
DBD6 ED 43 9D E0   1716   LD (#MSBT),BC ;DATA LENGTH
DBDA C9            1717   RET
DBDB               1718  ZBASAZS
DBDB 7C            1719   LD A,H ;
DBDC B5            1720   OR L   ;
DBDD C0            1721   RET NZ ;IF 上位16BITはNOT0
DBDE 18 F0         1722   JR ZBSAZ
DBE0               1723
DBE0               1724  ;(FILE-SIZE 初期化)
DBE0               1725  ;
DBE0               1726  FLSZCLR
DBE0 21 00 00      1727   LD HL,0
DBE3 22 86 D1      1728   LD (#FSZL),HL ;下位
DBE6 22 88 D1      1729   LD (#FSZH),HL ;上位
DBE9 CD 76 D7      1730   CALL ODST2
DBEC E5            1731   PUSH HL
DBED DD E1         1732   POP IX
DBEF DD 5E 10      1733   LD E,(IX+16)
DBF2 DD 56 11      1734   LD D,(IX+17)
DBF5 ED 53 F3 DD   1735   LD (SUBSL),DE ;WORK L
DBF9 DD 6E 12      1736   LD L,(IX+18)
DBFC DD 66 13      1737   LD H,(IX+19)
DBFF 22 F5 DD      1738   LD (SUBSH),HL ;WORK H
DC02 7A            1739   LD A,D
DC03 B3            1740   OR E
DC04 C0            1741   RET NZ
DC05 7C            1742   LD A,H
DC06 B5            1743   OR L
DC07 C0            1744   RET NZ
DC08               1745         ;SIZEが0の時
DC08               1746  ZBSAZ2 ;FATとSIZEが違う時
DC08 3A E5 EC      1747   LD A,(&ESCP) ;FAT優先
DC0B E6 0F         1748   AND $F        ;指定変更
DC0D 32 E5 EC      1749   LD (&ESCP),A
DC10 C9            1750   RET
DC11               1751
DC11               1752  FLSZSET ;SIZE更新
DC11 2A 86 D1      1753   LD HL,(#FSZL)
DC14 ED 5B 9D E0   1754   LD DE,(#MSBT)
DC18 19            1755   ADD HL,DE
DC19 22 86 D1      1756   LD (#FSZL),HL
DC1C 2A 88 D1      1757   LD HL,(#FSZH)
DC1F 11 00 00      1758   LD DE,0
DC22 ED 5A         1759   ADC HL,DE ;HL+0+繰り上がり
DC24 22 88 D1      1760   LD (#FSZH),HL
DC27 C9            1761   RET
DC28               1762
DC28               1763  ;/ DIR時のFILE-NAME DATA作成 /
DC28               1764  ;
DC28               1765  DIRSUB
DC28 AF            1766   XOR A
DC29 32 84 D1      1767   LD (#YEN),A
DC2C 2A A4 E0      1768   LD HL,(#BADR)
DC2F ED 5B 8F E0   1769   LD DE,(#BSIZ)
DC33 19            1770   ADD HL,DE
DC34 22 91 E0      1771   LD (#BF),HL ;DATA先頭ADR
DC37 EB            1772   EX DE,HL
DC38               1773  DRSLP
DC38 3A 8C E0      1774   LD A,(#STOP)
DC3B B7            1775   OR A
DC3C 20 11         1776   JR NZ,DRSLP2
DC3E CD 20 DD      1777   CALL BLDIR0 ;32バイト初期化
DC41 CD 81 DC      1778   CALL JITUHA ;作成
DC44 D5            1779   PUSH DE
DC45 CD 06 D0      1780   CALL #DIR2  ;NEXT検索
DC48 D1            1781   POP DE
DC49 21 84 D1      1782   LD HL,#YEN  ;
DC4C 34            1783   INC (HL)    ;FILE数
DC4D 18 E9         1784   JR DRSLP
DC4F               1785  DRSLP2
DC4F 3A 76 D0      1786   LD A,(#ALPHA+'W'-'A')
DC52 B7            1787   OR A
DC53 28 06         1788   JR Z,NOTWIDE
DC55 3E 0D         1789    LD A,$0D    ;WIDEで
DC57 CD 18 E0      1790    CALL #LDDEA ;改行
DC5A 13            1791    INC DE      ;
DC5B               1792  NOTWIDE
DC5B 3E 1A         1793   LD A,$1A     ;END-CODE
DC5D CD 18 E0      1794   CALL #LDDEA  ;
DC60 13            1795   INC DE       ;
DC61 2A 91 E0      1796   LD HL,(#BF)
DC64 EB            1797   EX DE,HL
DC65 B7            1798   OR A
DC66 ED 52         1799   SBC HL,DE
DC68 22 9D E0      1800   LD (#MSBT),HL
DC6B AF            1801   XOR A
DC6C 32 8C E0      1802   LD (#STOP),A
DC6F 3C            1803   INC A
DC70 32 99 E0      1804   LD (#EDW),A  ;END
DC73 32 9C E0      1805   LD (#IOFG),A ;1..出力要求
DC76 3C            1806   INC A
DC77 32 80 D1      1807   LD (#OD),A
DC7A FD 21 1C D6   1808   LD IY,NULRET
DC7E C3 BB D5      1809   JP DEVDRV
DC81               1810
DC81               1811  JITUHA ;OPTIONの振り分け
DC81 3A 76 D0      1812   LD A,(#ALPHA+'W'-'A')
DC84 B7            1813   OR A
DC85 20 0A         1814   JR NZ,WIDE ;IF WIDE
DC87 CD 98 DC      1815   CALL JITUHA2
DC8A 3E 0D         1816   LD A,$0D
DC8C CD 18 E0      1817   CALL #LDDEA
DC8F 13            1818   INC DE
DC90 C9            1819   RET
DC91               1820
DC91               1821  WIDE
DC91 CD 09 DD      1822   CALL TENSO ;NAMEのみ作成
DC94 13            1823   INC DE ;空白
DC95 13            1824   INC DE ;
DC96 13            1825   INC DE ;
DC97 C9            1826   RET
DC98               1827
DC98               1828  JITUHA2 ;実際の作成
DC98 CD 09 DD      1829   CALL TENSO ;13+3文字
DC9B 3A 8A E0      1830   LD A,(#FDOP+3)
DC9E B7            1831   OR A
DC9F 28 07         1832   JR Z,DEDDSKD ;IF NOT DIR
DCA1 21 06 00      1833    LD HL,6    ;空白
DCA4 19            1834    ADD HL,DE ;
DCA5 EB            1835    EX DE,HL  ;
DCA6 18 0E         1836    JR DEDDSK
DCA8               1837  DEDDSKD
DCA8 3A 7A E0      1838   LD A,(#FBYT+2)
DCAB 0E 00         1839   LD C,0
DCAD CD E4 DC      1840   CALL HEX2   ;SIZE H 1BYTE
DCB0 2A 78 E0      1841   LD HL,(#FBYT)
DCB3 CD D2 DC      1842   CALL HEXASC ;SIZE L 2BYTE
DCB6               1843  DEDDSK
DCB6 13            1844   INC DE
DCB7 CD 2D DD      1845   CALL OPTZ ;属性表示
DCBA 3A 87 E0      1846    LD A,(#FDOP)
DCBD E6 01         1847    AND 1
DCBF C8            1848    RET Z ;ASC,BAS
DCC0 3A 97 E0      1849    LD A,(#MAC)
DCC3 FE 02         1850    CP 2
DCC5 C8            1851    RET Z ;MS-DOS 2HD
DCC6 FE 04         1852    CP 4
DCC8 C8            1853    RET Z ;MS-DOS 2D
DCC9 13            1854   INC DE
DCCA 2A 83 E0      1855   LD HL,(#MMLD)
DCCD 0E 01         1856   LD C,1
DCCF C3 D2 DC      1857   JP HEXASC ;LOAD ADR
DCD2               1858
DCD2               1859  ;(2桁のHEX$(DATA))
DCD2               1860  ;
DCD2               1861  HEXASC
DCD2 7C            1862   LD A,H
DCD3 CD E4 DC      1863   CALL HEX2 ;1桁分
DCD6 7D            1864   LD A,L
DCD7 CD E4 DC      1865   CALL HEX2 ;
DCDA 79            1866   LD A,C ;C=0?のFLUG
DCDB B7            1867   OR A
DCDC C0            1868   RET NZ ;IF DATA<>0
DCDD 1B            1869   DEC DE
```

▶へえっ、へえーだ。まじめに働くと月に８万円もたまるんだゼー。８カ月も働きゃ、おめー、64万円だ。Macが買えるんだよー。　　中井　猛 (18) 徳島県

```
DCDE 3E 30          1870   LD A,'0'     ;0を表示
DCE0 CD 18 E0       1871   CALL #LDDEA ;
DCE3 13             1872   INC DE
DCE4                1873  RET
DCE4                1874
DCE4                1875  HEX2 ;IN A
DCE4 F5             1876   PUSH AF
DCE5 E6 F0          1877   AND $F0 ;4BIT分抽出
DCE7 0F             1878   RRCA
DCE8 0F             1879   RRCA
DCE9 0F             1880   RRCA
DCEA 0F             1881   RRCA
DCEB CD F1 DC       1882   CALL HEXSB ;4BIT分
DCEE F1             1883   POP AF
DCEF E6 0F          1884   AND $F ;下位4BIT
DCF1                1885  HEXSB
DCF1 B7             1886   OR A
DCF2 28 01          1887   JR Z,HEXSB2
DCF4 0C             1888    INC C ;C=NOT 0 FLUG
DCF5                1889  HEXSB2
DCF5 C6 30          1890   ADD A,'0'
DCF7 FE 3A          1891   CP '9'+1
DCF9 38 02          1892   JR C,HEXSK1 ;IF 数字
DCFB C6 07          1893    ADD A,'A'-'9'-1
DCFD                1894  HEXSK1
DCFD CD 18 E0       1895   CALL #LDDEA
DD00 79             1896   LD A,C    ;IF 0 なら
DD01 B7             1897   OR A      ;0じゃなくて
DD02 3E 20          1898   LD A,' ' ;' 'を出力
DD04 CC 18 E0       1899   CALL Z,#LDDEA
DD07 13             1900   INC DE
DD08 C9             1901   RET
DD09                1902
DD09                1903  ;(DATA SET)
DD09                1904  ;
DD09                1905  TENSO
DD09 21 68 E0       1906   LD HL,#FNAM ;NAME
DD0C 06 0D          1907   LD B,13      ;13文字
DD0E CD 17 DD       1908   CALL BLDIR  ;
DD11 13             1909   INC DE
DD12 06 03          1910   LD B,3       ;3文字
DD14 C3 17 DD       1911   JP BLDIR ;しまった!
DD17                1912  BLDIR
DD17 7E             1913   LD A,(HL)
DD18 23             1914   INC HL
DD19 CD 18 E0       1915   CALL #LDDEA
DD1C 13             1916   INC DE
DD1D 10 F8          1917   DJNZ BLDIR
DD1F C9             1918   RET
DD20                1919  BLDIR0 ;32バイトCLEAR
DD20 D5             1920   PUSH DE
DD21 06 20          1921   LD B,32
DD23                1922  BLDIRLP
DD23 3E 20          1923   LD A,' '
DD25 CD 18 E0       1924   CALL #LDDEA
DD28 13             1925   INC DE
DD29 10 F8          1926   DJNZ BLDIRLP
DD2B D1             1927   POP DE
DD2C C9             1928   RET
DD2D                1929
DD2D                1930  ;(属性の表示用DATA SET)
DD2D                1931  ;
DD2D                1932  OPTZ
DD2D 3A 89 E0       1933   LD A,(#FDOP+2)
DD30 B7             1934   OR A
DD31 3E 50          1935   LD A,'P'
DD33 28 03          1936   JR Z,OPTZ2
DD35 CD 18 E0       1937    CALL #LDDEA ;'P' SET
DD38                1938  OPTZ2
DD38 13             1939   INC DE
DD39 3A 8A E0       1940   LD A,(#FDOP+3)
DD3C B7             1941   OR A
DD3D 21 C2 DD       1942   LD HL,ASCDT
DD40 20 0B          1943   JR NZ,OPTZ3 ;IF DIR
DD42 3A 87 E0       1944    LD A,(#FDOP)
DD45 06 04          1945    LD B,4
DD47                1946  OPTZLP ;Bin,Bas,Ascの判定
DD47 23             1947    INC HL
DD48 0F             1948    RRCA
DD49 38 02          1949    JR C,OPTZ3
DD4B 10 FA          1950    DJNZ OPTZLP
DD4D                1951  OPTZ3
DD4D 7E             1952   LD A,(HL) ;(ASCDT+?)
DD4E CD 18 E0       1953   CALL #LDDEA
DD51 13             1954   INC DE
DD52 C9             1955   RET
DD53                1956
DD53                1957  ;(FORMAT別DATA-INDEXのSET)
DD53                1958  ;
DD53                1959  SETFNDT
DD53 3A 97 E0       1960   LD A,(#MAC)
DD56 DD 21 3C E1    1961   LD IX,FNDT1 ;X12HD
DD5A 3D             1962   DEC A
DD5B C8             1963   RET Z
DD5C DD 21 44 E1    1964   LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D             1965   DEC A
DD61 C8             1966   RET Z
DD62 DD 21 3C E1    1967   LD IX,FNDT1 ;X12D
DD66 3D             1968   DEC A
DD67 C8             1969   RET Z
DD68 DD 21 44 E1    1970   LD IX,FNDT2 ;MS-DOS 2D
DD6C C9             1971   RET
DD6D                1972
DD6D                1973  ;/ 256バイト転送 /
DD6D                1974  ;
DD6D                1975  TRS256 ;IN BF OUT BF
DD6D 06 00          1976   LD B,0
DD6F 2A 91 E0       1977   LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD       1978   CALL TRSB
DD75 22 91 E0       1979   LD (#BF),HL
DD78 C9             1980   RET
DD79                1981
DD79                1982  TRSB
DD79 11 00 EE       1983   LD DE,P256 ;P256へ
DD7C                1984  TRSLP256
DD7C CD 12 E0       1985    CALL #LDAHL
DD7F 23             1986    INC HL
DD80 12             1987    LD (DE),A
DD81 13             1988    INC DE
DD82 10 F8          1989    DJNZ TRSLP256
DD84 C9             1990   RET
DD85                1991
DD85                1992  ;/ ドライブ番号の判定 /
DD85                1993  ;
DD85                1994  SETDN ;IN B,(DE)
DD85 78             1995   LD A,B
DD86 B7             1996   OR A
DD87 C8             1997   RET Z ;IF DATA=0
DD88 13             1998   INC DE
DD89 1A             1999   LD A,(DE)
DD8A 1B             2000   DEC DE
DD8B FE 3A          2001   CP ':' ;X:?
DD8D 20 27          2002   JR NZ,SETDNRT2
DD8F 1A             2003   LD A,(DE)
DD90 FE 61          2004   CP 'a'
DD92 38 02          2005   JR C,SETDNSK
DD94 D6 20          2006    SUB 'a'-'A'
DD96                2007  SETDNSK
DD96 D6 41          2008   SUB 'A'
DD98 FE 1A          2009   CP 'Z'-'A'+1
DD9A 38 07          2010   JR C,SETDNRT ;IF 正しい
DD9C 3E 04          2011   LD A,4
DD9E 32 BC E0       2012   LD (#STOP),A
DDA1 37             2013   SCF ;エラー
DDA2 C9             2014   RET
DDA3                2015  SETDNRT
DDA3 32 8B E0       2016   LD (#DN),A  ;SET
DDA6 21 C0 EC       2017   LD HL,&MAC4
DDA9 C5             2018   PUSH BC
DDAA CD B8 DD       2019   CALL DNST2
DDAD C1             2020   POP BC
DDAE 7E             2021   LD A,(HL)
DDAF 32 97 E0       2022   LD (#MAC),A ;FORMAT SET
DDB2 13             2023   INC DE ;
DDB3 13             2024   INC DE ;POINTERを
DDB4 05             2025   DEC B  ;'X:'の次へ
DDB5 05             2026   DEC B  ;
DDB6                2027  SETDNRT2
DDB6 B7             2028   OR A
DDB7 C9             2029   RET
DDB8                2030
DDB8                2031  DNST2 ;HL=HL+(#DN)
DDB8 C5             2032   PUSH BC
DDB9 06 00          2033   LD B,0
DDBB 3A 8B E0       2034   LD A,(#DN)
DDBE 4F             2035   LD C,A
DDBF 09             2036   ADD HL,BC
DDC0 C1             2037   POP BC
DDC1 C9             2038   RET
DDC2                2039
DDC2                2040  ;/ WORK エリア 2 /
DDC2                2041  ;
DDC2 44 4D 42 41    2042  ASCDT DB 'D','M','B','A',' '
DDC6 20
DDC7                2043  BGAM  DS 35
DDEA 54 E0          2044  BFNM  DW #FLNM
DDEC 00 00          2045  FREE  DW 0
DDEE 00             2046  NXTDB DB 0
DDEF 00 00          2047  BFA   DW 0
DDF1 00 00          2048  BSA   DW 0
DDF3 00 00          2049  SUBSL DW 0
DDF5 00 00          2050  SUBSH DW 0
DDF7 00             2051  SMAC  DB 0
DDF8 00 00          2052  DDUSH DW 0
DDFA                2053
DDFA                2054  ;/ 固定データ /
DDFA                2055  ;
DDFA                2056  DISK0 ;表示用
DDFA 30 3A          2057   DM '0:'
DDFC 31 3A          2058   DM '1:'
DDFE 32 3A          2059   DM '2:'
DE00 33 3A          2060   DM '3:'
DE02                2061  MEM0:
DE02 4D 45 4D 30    2062   DM 'MEM0:'
DE06 3A
DE07 4D 45 4D 31    2063   DM 'MEM1:'
DE0B 3A
DE0C 45 4D 4D 30    2064   DM 'EMM0:'
DE10 3A
DE11 45 4D 4D 31    2065   DM 'EMM1:'
DE15 3A
DE16 45 4D 4D 32    2066   DM 'EMM2:'
DE1A 3A
DE1B 45 4D 4D 33    2067   DM 'EMM3:'
DE1F 3A
DE20                2068  SHIME
DE20 00 00          2069   DW 0
DE22                2070
DE22                2071  ;
E1BF 10
E190                2240
E190                2241  ;
E190                2242  ;
EC40                2243   ORG $EC40
EC40                2244  SRDR
EC40 4E 55 4C       2245   DM 'NUL'  ;0
EC43 01             2246   DB 1
EC44 1C D6          2247   DW NULRET
EC46 43 4F 4E       2248   DM 'CON'  ;1
EC49 00             2249   DB 0
EC4A 00 00          2250   DW 0
EC4C 41 55 58       2251   DM 'AUX'  ;2
EC4F 00             2252   DB 0
EC50 00 00          2253   DW 0
EC52 50 52 4E       2254   DM 'PRN'  ;3
EC55 00             2255   DB 0
EC56 00 00          2256   DW 0
EC58 24 31 40       2257   DM '$1@'  ;4
EC5B 00             2258   DB 0
EC5C 00 00          2259   DW 0
EC5E 24 32 40       2260   DM '$2@'  ;5
EC61 00             2261   DB 0
EC62 00 00          2262   DW 0
EC64 24 33 40       2263   DM '$3@'  ;6
EC67 00             2264   DB 0
EC68 00 00          2265   DW 0
EC6A 00 00 00       2266   DB 0,0,0  ;7
EC6D 01             2267   DB 1
EC6E DD D5          2268   DW DEVOI
EC70                2269
EC70                2270  ;/ PATH DATA /
EC70                2271  ;
EC80                2272   ORG $EC80
EC80                2273  PATH
EC80                2274
```

リスト3

```
0000                   1  ;////////////////////////////
0000                   2  ; INTEGRAL X  No.2
0000                   3  ;  (KAME-DOS) for REDA
0000                   4  ; COPYRIGHT 1990 M.KAMEDA
0000                   5  ;////////////////////////////
0000                   6  ;
0000                   7
0000                   8  ;注)プログラムの一部で
0000                   9  ;     祝 一平著
0000                  10  ;    「試験に出るX1」
0000                  11  ;     を参考にさせて
0000                  12  ;     いただきました。
0000                  13
EE00 P                14  P256  EQU $EE00 ;256 BUFF
ECC0 P                15  ST    EQU $ECC0
ECC0 P                16  &MAC4 EQU ST     ;FORMAT
ECDA P                17  &TR4  EQU ST+26 ;TRACK
ECDE P                18  &DN   EQU ST+30 ;DRIVE
ECDF P                19  &BUFF EQU ST+31 ;BUFF ADR
ECE1 P                20  &FF   EQU ST+33 ;FAT ADR
ECE3 P                21  &BSIZ EQU ST+35 ;BUFF LENGTH
ECE5 P                22  &ESCP EQU ST+37 ;EOF FLUG
ECE6 P                23  &IOMM EQU ST+38 ;BUFF 位置
ECE7 P                24  &EOF2 EQU ST+39 ;EOF
ECE8 P                25  &EOF3 EQU ST+40 ;変更後EOF
0000                  26
0000                  27  ;/* D-IOCS部分 */
0000                  28
E000                  29   ORG $E000
E000                  30
E000 C3 A6 E7         31  #RWREC JP RWREC
E003 C3 61 E5         32  #DLFAT JP DELFAT
E006 C3 F2 E5         33  #FATWT JP FATWT
E009 C3 A1 E2         34  #CRSRW JP CRSRW
E00C C3 00 E2         35  #DEVIC JP DEVICE
E00F C3 90 E6         36  #LDHLA JP LDHLA
E012 C3 7A E6         37  #LDAHL JP LDAHL
E015 C3 8A E6         38  #LDADE JP LDADE
E018 C3 96 E6         39  #LDDEA JP LDDEA
E01B C3 82 E5         40  #VAR   JP VAR
E01E C3 3F E8         41  #WTTRC JP WTTRC
E021 C3 6A E4         42  #SEC00 JP SEC00
E024 C3 28 E7         43  #RSTOR JP RSTOR
E027 C3 63 E6         44  #MLTHD JP MLTHD
E02A C3 44 E6         45  #DIVHD JP DIVHD
E02D C3 B3 E6         46  #BGIOM JP BGIOMM
E030 C3 D9 E6         47  #BGRST JP BGRST
E033 C3 58 EA         48  #ERR0  JP ERR0
E036 C3 65 EA         49  #ERR3  JP ERR3
E039 C3 69 EA         50  #ERR4  JP ERR4
E03C C3 6D EA         51  #ERR5  JP ERR5
E03F C3 71 EA         52  #ERR6  JP ERR6
E042 C3 75 EA         53  #ERR7  JP ERR7
E045 C3 94 EA         54  #GYOKI JP GYOKI
E048 C3 D8 EA         55  #SPATH JP SPATH
E04B                  56
E051                  57   ORG $E051
E051                  58
E051                  59  ;  USER
E051 00               60  #DIRN DB 0 ;階層の深さ
E052 00 00            61  #MORF DW 0 ;DIR ADR
E054 43 4F 4E 46      62  #FLNM DM "CONFI" ;FILE-
E058 49
E059 47 20 20 20      63        DM "G    "  ;NAME
E05D 20 20 20 20      64        DM "    "   ;
E061 53 59 53         65        DM "SYS"    ;
E064 00               66  #DPNF DB 0 ;HIDE
E065 00               67  #SBDR DB 0 ;下位DIR
E066 00               68  #WTNF DB 0 ;書き込み禁止
E067 00               69  #BBAF DB 0 ;FILEの種類
E068                  70  ;  DISK
E068                  71  #FNAM  DS 13 ;FILE-NAME
E075                  72  #EXTN  DS 3  ;拡張子
E078                  73  #FBYT  DS 4  ;FILE-SIZE
E07C                  74  #ECRS  DS 2  ;クラスター
E07E                  75  #DATE  DS 5  ;日付
E083                  76  #MMLD  DS 2  ;LOAD ADR
E085                  77  #MMJP  DS 2  ;JUMP ADR
E087                  78  #FDOP  DS 4 ;3,$10,$40,$80(X1)
E08B                  79
E08B 01               80  #DN   DB 1      ;DRIVE
E08C 00               81  #STOP DB 0      ;エラー
E08D 00 C0            82  #FF   DW $C000 ;FAT ADR
E08F 00 10            83  #BSIZ DW $1000 ;BUFF LENGTH
E091 00 00            84  #BF   DW 0      ;BUFF ADR
E093 00 00            85  #REC  DW 0      ;レコード
E095 0B 00            86  #CRS  DW 11     ;クラスター
E097 02               87  #MAC  DB 2      ;FORMAT
E098 20               88  #PASS DB 32     ;パスワード
E099 00               89  #EDW  DB 0      ;END OR WRITE
E09A 00               90  #IOMM DB 0      ;BUFF位置
E09B 00               91  #EDR  DB 0      ;END OF READ
E09C 00               92  #IOFG DB 0      ;入出力FLUG
E09D 00 10            93  #MSBT DW $1000 ;
E09F 00 00            94  #SLNG DW 0      ;SECTORの長さ
E0A1 00               95  #FRWF DB 0      ;R & W FLUG
E0A2 00               96  #WFD0 DB 0      ;I/O($1FD0)
E0A3 00               97  #ZOKU DB 0      ;属性
E0A4 00 C0            98  #BADR DW $C000 ;BUFF ADR
E0A6 00               99  #RSCMD DB 0     ;FDC
E0A7 18              100  #SKCMD1 DB $18 ;
E0A8 1C              101  #SKCMD2 DB $1C ;
E0A9 00 00           102  #BPATH DW 0     ;PATH ADR
E0AB 01              103  #SMACS DB 1     ;#MAC WORK
E0AC 0E              104  #WAIT  DB 14    ;
E0AD                 105
E100                 106   ORG $E100
E100                 107
E100                 108  ;/ FIX DATA /
E100                 109  ;
E100                 110  DVCDT1 ;X12HD
E100 00 01           111   DW 256    ;SECTOR
E102 1A 10           112   DB 26,16 ;
E104 00 10 1C 00     113   DW 4096,28,29
E108 1D 00
E10A 50              114   DB 80     ;LAST TRACK
E10B A3 0F           115   DW 4003   ;LAST SEC
E10D                 116  DVCDT2 ;MS-DOS2HD
E10D 00 04           117   DW 1024
E10F 08 01           118   DB 8,1
E111 00 04 01 00     119   DW 1024,1,2
E115 02 00
E117 50              120   DB 80
E118 CF 04           121   DW $4CF
E11A                 122  DVCDT3 ;X12D
E11A 00 01           123   DW 256
E11C 10 10           124   DB 16,16
E11E 00 10 0E 00     125   DW 4096,14,14
E122 0E 00
E124 28              126   DB 40
E125 FF 04           127   DW 1279
E127                 128  DVCDT4 ;MS-DOS2D
E127 00 02           129   DW 512
E129 09 02           130   DB 9,2
E12B 00 04 01 00     131   DW 1024,1,2
E12F 02 00
E131 28              132   DB 40
E132 CF 02           133   DW 719
E134                 134
E134                 135  RDDT1
E134 20 10           136   DB 32,16 ;ROOT DIR
E136                 137  RDDT2       ;レコード
E136 05 06           138   DB 5,6    ;番号と
E138                 139  RDDT3       ;レコード
E138 10 10           140   DB 16,16 ;数
E13A                 141  RDDT4
E13A 05 07           142   DB 5,7
E13C                 143
E13C                 144  FNDT1 ;DIR 情報 X1
E13C FF 00 01 0D     145   DB $FF,0,1,13,0,$10,0,$80
E140 00 10 00 80
E144                 146  FNDT2 ;MS-DOS
E144 00 E5 00 08     147   DB 0,$E5,0,8,11,$2,8,$10
E148 0B 02 08 10
E14C                 148
E14C                 149  NADT1 ;DIR 情報2 X1
E14C 08              150   DB 8
E14D 01 0D           151   DB 1,13
E14F 68 E0           152   DW #FNAM
E151 0E 03           153   DB 14,3
E153 75 E0           154   DW #EXTN
E155 11 01           155   DB 17,1
E157 98 E0           156   DW #PASS
E159 12 02           157   DB 18,2
E15B 78 E0           158   DW #FBYT
E15D 1E 02           159   DB 30,2
E15F 7C E0           160   DW #ECRS
E161 18 05           161   DB 24,5
E163 7E E0           162   DW #DATE
E165 14 02           163   DB 20,2
E167 83 E0           164   DW #MMLD
E169 16 02           165   DB 22,2
E16B 85 E0           166   DW #MMJP
E16D 00 07 10 40     167   DB 0, 7,$10,$40,$80
```

▶某月某日　キーボードにお茶をかけてしまった！　キーボードをばらして乾かして，5日後に回復！　ソフトキーボードってこの日のためにあったんだ（納得）。「糖分を含んだものであれば，水洗いしてから乾かしたほうがよい」とM氏がいっていた……。フーン，みんな結構やっているんだね。
中村　マサエ（31）東京都

```
E171 80
E172                168  NADT2 ;MS-DOS
E172 06             169   DB 6
E173 00 08          170   DB 0,8
E175 68 E0          171   DW #FNAM
E177 20 05          172   DB 32,5
E179 70 E0          173   DW #FNAM+8
E17B 08 03          174   DB 8,3
E17D 75 E0          175   DW #EXTN
E17F 1C 04          176   DB 28,4
E181 78 E0          177   DW #FBYT
E183 1A 02          178   DB 26,2
E185 7C E0          179   DW #ECRS
E187 16 04          180   DB 22,4
E189 7E E0          181   DW #DATE
E18B 0B 00 02 01    182   DB 11, 0,2,1,$10
E18F 10
E190                183
E200                184   ORG $E200
E200                185
E200                186  ;/ PROGRAM /
E200                187  ;
E200                188  ;( FORMAT判定とFAT読み込み)
E200                189  ;
E200                190  DEVICE ;IN BSIZ,FF
E200 C5             191   PUSH BC
E201 D5             192   PUSH DE
E202 E5             193   PUSH HL
E203 AF             194   XOR A ;READ
E204 32 A1 E0       195   LD (#FRWF),A
E207 CD 1A E2       196   CALL FRDVC ;自動判断
E20A 38 0A          197   JR C,DNDNST
E20C 21 C0 EC       198   LD HL,&MAC4
E20F CD EE E6       199   CALL DNST
E212 3A 97 E0       200   LD A,(#MAC)
E215 77             201   LD (HL),A ;FORMAT 変更
E216                202  DNDNST
E216 E1             203   POP HL
E217 D1             204   POP DE
E218 C1             205   POP BC
E219 C9             206   RET
E21A                207
E21A                208  FRDVC
E21A 3A 8B E0       209    LD A,(#DN)
E21D FE 04          210    CP 4
E21F D2 81 E2       211    JP NC,RMDVC ;RAM-DISK
E222                212
E222 CD 28 E7       213   CALL RSTOR ;RESTOR
E225 D8             214   RET C
E226 CD 1B E0       215   CALL #VAR  ;DATA SET
E229 CD 21 E6       216   CALL FATRW ;FAT READ
E22C 30 30          217   JR NC,DVCPAT ;IF エラーなし
E22E                218  DVCMIS
E22E 3A AB E0       219   LD A,(#SMACS) ;最初のMAC
E231 32 97 E0       220   LD (#MAC),A
E234                221  DVCLP1
E234 CD 33 E0       222    CALL #ERR0
E237 CD FC E6       223     CALL MTRON ;MOTOR ON
E23A 3E 02          224     LD A,2 ;0から2へHEAD移動
E23C 32 80 EB       225     LD (TRN),A
E23F D4 4D E7       226     CALL NC,SEEKT ;SEEK
E242 D8             227     RET C
E243 CD 24 E0       228    CALL #RSTOR ;RSTOR
E246 CD 1B E0       229    CALL #VAR   ;DATA SET
E249 CD 21 E6       230    CALL FATRW  ;FAT READ
E24C D0             231    RET NC
E24D 21 97 E0       232    LD HL,#MAC  ;
E250 34             233    INC (HL)    ;FORMAT
E251 3E 04          234    LD A,4      ;
E253 BE             235    CP (HL).    ;4まで
E254 30 DE          236    JR NC,DVCLP1
E256 3E 03          237    LD A,3
E258 32 97 E0       238    LD (#MAC),A
E25B C3 61 EA       239   JP ERR2 ;FAULT エラー
E25E                240
E25E                241  DVCPAT ;最終CHECK
E25E 3A 97 E0       242   LD A,(#MAC)
E261 0E FE          243   LD C,$FE ;C=FATの先頭
E263 FE 02          244   CP 2
E265 28 06          245   JR Z,DVCSK
E267 0E FD          246   LD C,$FD
E269 FE 04          247   CP 4
E26B 20 12          248   JR NZ,DVCRT2
E26D                249  DVCSK
E26D 2A 8D E0       250   LD HL,(#FF) ;MAC=MSDOS
E270 CD 7A E6       251   CALL LDAHL
E273 B9             252   CP C
E274 20 B8          253   JR NZ,DVCMIS ;IF 違う
E276 23             254   INC HL
E277 CD 7A E6       255   CALL LDAHL
E27A 2B             256   DEC HL
E27B FE FF          257   CP $FF
E27D 20 AF          258   JR NZ,DVCMIS
E27F                259  DVCRT2 ;最終CHECK OK
E27F B7             260   OR A
E280 C9             261   RET
E281                262
E281                263  ;
E281                264  RMDVC ;RAM-DISK
E281 3E 03          265   LD A,3 ;ONLY X1
E283 32 97 E0       266   LD (#MAC),A
E286 CD 1B E0       267   CALL #VAR  ;DATA SET
E289 CD 21 E6       268   CALL FATRW ;FAT READ
E28C 2A 8D E0       269   LD HL,(#FF)
E28F CD 7A E6       270   CALL LDAHL
E292 FE 01          271   CP $01 ;FATの先頭
E294 20 08          272   JR NZ,RMDVCDM
E296 23             273   INC HL
E297 CD 7A E6       274   CALL LDAHL
E29A FE 8F          275   CP $8F ;次の1バイト
E29C 28 E1          276   JR Z,DVCRT2
E29E                277  RMDVCDM ;エラー
E29E C3 61 EA       278   JP ERR2
E2A1                279
E2A1                280  ;( クラスター読み書き)
E2A1                281  ;
E2A1                282  CRSRW ;IN BF,FF,BSCS,CRS OUT MS
                         BT
E2A1 C5             283   PUSH BC
E2A2 D5             284   PUSH DE
E2A3 E5             285   PUSH HL
E2A4 3A 8B E0       286    LD A,(#DN)
E2A7 FE 04          287    CP 4
E2A9 30 1E          288    JR NC,CRSRRS ;RAM-DISK
E2AB CD FC E6       289   CALL MTRON ;MOTOR ON
E2AE 38 1C          290   JR C,CRSRRT
E2B0 21 DA EC       291   LD HL,&TR4 ;特殊処理
E2B3 CD EE E6       292   CALL DNST  ;よくは
E2B6 3A 87 EB       293   LD A,(LTR) ;わからないが
E2B9 4F             294   LD C,A     ;2TRACK以上
E2BA 7E             295   LD A,(HL)  ;動かさないと
E2BB C6 02          296   ADD A,2    ;ダメなことが
E2BD B9             297   CP C       ;ある
E2BE 38 01          298   JR C,CRSRRSK
E2C0 97             299    SUB A,4
E2C1                300  CRSRRSK     ;
E2C1 32 80 EB       301   LD (TRN),A ;
E2C4 CD 4D E7       302   CALL SEEKT ;SEEK
E2C7 38 03          303   JR C,CRSRRT
E2C9                304  CRSRRS
E2C9 CD D0 E2       305   CALL CRSRW2
E2CC                306  CRSRRT
E2CC E1             307   POP HL
E2CD D1             308   POP DE
E2CE C1             309   POP BC
E2CF C9             310   RET
E2D0                311
E2D0                312  ;BUFFの大きさだけ
E2D0                313  ; クラスターを
E2D0                314  ; を読み書きする
E2D0                315  CRSRW2
E2D0 AF             316   XOR A
E2D1 32 92 EB       317   LD (RCN),A  ;レコードの数
E2D4 3A 8A EB       318   LD A,(BSCS) ;クラスター数
E2D7 32 8B EB       319   LD (BSCSWK),A
E2DA                320  CSRLP1
E2DA B7             321   OR A ;(BSCS)回処理する
E2DB 28 2D          322   JR Z,CSRDED2
E2DD CD 20 E3       323   CALL CTOR     ;クラスターか
E2E0 CD 56 E3       324   CALL SECHFAT ;らレコードを
E2E3 38 35          325   JR C,CSRDED  ;計算 & FAT
E2E5 3A 95 EB       326   LD A,(RIC) ;レコードIN
E2E8 47             327   LD B,A     ;クラスター
E2E9 3A 96 EB       328   LD A,(ED)  ;END OF FAT
E2EC B7             329   OR A
E2ED 28 01          330   JR Z,FILED ;IF END
E2EF 47             331    LD B,A
E2F0                332  FILED
E2F0                333  RDRCLP
E2F0 CD 00 E0       334   CALL #RWREC ;レコード
E2F3 38 25          335   JR C,CSRDED
E2F5 21 92 EB       336   LD HL,RCN
E2F8 34             337   INC (HL) ;レコード数
E2F9 10 F5          338   DJNZ RDRCLP
E2FB 3A 96 EB       339    LD A,(ED)
E2FE B7             340    OR A
E2FF 20 09          341    JR NZ,CSRDED2
E301 3A 8B EB       342   LD A,(BSCSWK)
E304 3D             343   DEC A
E305 32 8B EB       344   LD (BSCSWK),A
E308 18 D0          345   JR CSRLP1
E30A                346  CSRDED2 ;BUFF満杯
E30A 3A 92 EB       347   LD A,(RCN)
E30D 6F             348   LD L,A
E30E 26 00          349   LD H,0
E310 ED 5B 9F E0    350   LD DE,(#SLNG)
E314 CD 63 E6       351   CALL MLTHD
E317 22 9D E0       352   LD (#MSBT),HL ;DATAの長さ
E31A                353  CSRDED
E31A 3E 00          354   LD A,0
E31C 32 96 EB       355   LD (ED),A
E31F C9             356   RET
E320                357
E320                358  ;( クラスターから
E320                359  ;    レコードを計算する)
E320                360  ;
E320                361  CTOR ;IN CRS,MAC OUT REC
E320 2A 95 E0       362   LD HL,(#CRS) ;クラスター
E323 3A 97 E0       363   LD A,(#MAC)  ;FORMAT
E326 FE 02          364   CP 2
E328 28 1F          365    JR Z,CRNZ2
E32A FE 04          366   CP 4
E32C 28 21          367    JR Z,CRNZ4
E32E 11 00 01       368   LD DE,$100
E331 B7             369   OR A
E332 ED 52          370   SBC HL,DE
E334 38 09          371   JR C,CRB100 ;IF CRS<$100
E336 19             372   ADD HL,DE ;HL=クラスター
E337 11 80 00       373   LD DE,128 ;HL=HL-128
E33A B7             374   OR A      ;
E33B ED 52          375   SBC HL,DE ;
E33D 18 01          376   JR CRB101
E33F                377  CRB100
E33F 19             378   ADD HL,DE ;HL=クラスター
E340                379  CRB101
E340 29             380   ADD HL,HL ;
E341 29             381   ADD HL,HL ;
E342 29             382   ADD HL,HL ;
E343 29             383   ADD HL,HL ;HL=HL*16
E344                384  CRB110
E344 22 93 E0       385   LD (#REC),HL ;レコード
E347 B7             386   OR A
E348 C9             387   RET
E349                388
E349                389  CRNZ2 ;MS-DOS 2HD
E349 11 09 00       390   LD DE,9   ;
E34C 19             391   ADD HL,DE ;HL=HL+9
E34D 18 F5          392   JR CRB110
E34F                393  CRNZ4 ;MS-DOS 2D
E34F 29             394   ADD HL,HL ;
E350 11 08 00       395   LD DE,8   ;
E353 19             396   ADD HL,DE ;HL=HL*2+8
E354 18 EE          397   JR CRB110
E356                398
E356                399  ;( 次のFATをさがす )
E356                400  ;
E356                401  SECHFAT ;IN FF,CRS,MAC,MSBT OUT
                         CRS,ED
E356 3A A1 E0       402    LD A,(#FRWF)
E359 B7             403    OR A
E35A C2 75 E3       404    JP NZ,WTSFAT ;WRITE
E35D AF             405   XOR A
E35E 32 96 EB       406   LD (ED),A
E361 32 9B E0       407   LD (#EDR),A
E364 CD AE E3       408   CALL SECFAT2
E367 3A 8F EB       409   LD A,(EDD)  ;END OF ?
E36A 32 96 EB       410   LD (ED),A   ;FLUG
E36D 32 9B E0       411   LD (#EDR),A ;
E370 32 99 E0       412   LD (#EDW),A ;
E373 B7             413   OR A
E374                414  WTSECRT
E374 C9             415   RET
E375                416
E375                417  ;未使用FATをさがして
E375                418  ;書き込む
E375                419  WTSFAT ;IN EDW,MSBT
E375 2A 9D E0       420   LD HL,(#MSBT)
E378 ED 5B 93 EB    421   LD DE,(CSBYT) ;クラスター
E37C B7             422   OR A
E37D ED 52          423   SBC HL,DE
E37F 22 9D E0       424   LD (#MSBT),HL
E382 28 02          425   JR Z,WTSFAT89 ;C&Z THEN
E384 30 1B          426   JR NC,WTSFAT0 ;JR WTSFAT89
E386                427  WTSFAT89
E386 3A 99 E0       428    LD A,(#EDW)
E389 B7             429    OR A
E38A 28 15          430    JR Z,WTSFAT0 ;IF NOT END
E38C AF             431    XOR A
E38D 32 99 E0       432    LD (#EDW),A
E390 19             433    ADD HL,DE ;HL=(#MSBT)
E391 ED 5B 9F E0    434    LD DE,(#SLNG)
E395 CD 44 E6       435    CALL DIVHD
E398 7C             436    LD A,H
E399 B5             437    OR L  ;余り=0?
E39A 28 01          438    JR Z,WTSFSK90
E39C 1C             439     INC E
E39D                440  WTSFSK90
E39D 7B             441    LD A,E ;END FLUG
E39E 32 96 EB       442    LD (ED),A
E3A1                443  WTSFAT0
E3A1 2A 95 E0       444   LD HL,(#CRS)
E3A4 22 90 EB       445   LD (CRSS),HL
E3A7 CD 21 E0       446   CALL #SEC00 ;未使用をさがせ
E3AA D8             447   RET C
E3AB C3 DB E4       448   JP WTFAT2 ;(CRSS)<-CRS
E3AE                449
E3AE                450  ;次のFATをさがせ
E3AE                451  ;
E3AE                452  SECFAT2 ;IN FF,CRS,MAC OUT CR
                         S,EDD
E3AE AF             453   XOR A
E3AF 32 8F EB       454   LD (EDD),A
E3B2 3A 97 E0       455   LD A,(#MAC)
E3B5 3D             456   DEC A
E3B6 28 09          457   JR Z,SFAT1
E3B8 3D             458   DEC A
E3B9 28 33          459   JR Z,SFAT2
E3BB 3D             460   DEC A
E3BC CA 4F E4       461   JP Z,SFAT3
E3BF 18 2D          462   JR SFAT2
E3C1                463
E3C1                464  SFAT1 ;X12HD
E3C1 2A 8D E0       465   LD HL,(#FF)  ;FAT ADR
E3C4 ED 4B 95 E0    466   LD BC,(#CRS) ;FATの位置
E3C8 09             467   ADD HL,BC
E3C9 CD 7A E6       468   CALL LDAHL
E3CC 5F             469   LD E,A ;
E3CD 01 80 00       470   LD BC,128 ;上位は
E3D0 09             471   ADD HL,BC ;(+128バイト)
E3D1 CD 7A E6       472   CALL LDAHL
E3D4 57             473   LD D,A ;DE=次の位置
E3D5 ED 53 95 E0    474   LD (#CRS),DE
E3D9 21 7F 00       475   LD HL,$7F
E3DC B7             476   OR A
E3DD ED 52          477   SBC HL,DE
E3DF D0             478   RET NC ;IF $7F以下
E3E0 21 8F 00       479   LD HL,$8F
E3E3 B7             480   OR A
E3E4 ED 52          481   SBC HL,DE
E3E6 D8             482   RET C  ;IF $8F以上
E3E7 7B             483   LD A,E
E3E8 D6 7F          484   SUB $7F
E3EA 32 8F EB       485   LD (EDD),A ;使用RECORD
E3ED C9             486   RET
E3EE                487
E3EE                488  SFAT2 ;MS-DOS 2HD
E3EE 2A 95 E0       489   LD HL,(#CRS)
E3F1 CB 3C          490   SRL H ;
E3F3 CB 1D          491   RR L  ;HL=HL/2
E3F5 08             492   EX AF,AF' ;C FLUG
E3F6 E5             493   PUSH HL
E3F7 29             494   ADD HL,HL ;
E3F8 D1             495   POP DE    ;
E3F9 19             496   ADD HL,DE ;HL=(#CRS)/2*3
E3FA EB             497   EX DE,HL
E3FB 2A 8D E0       498   LD HL,(#FF) ;FAT ADR
E3FE 19             499   ADD HL,DE ;
E3FF 08             500   EX AF,AF'  ;C FLUG=(#CRS)
E400 38 0D          501   JR C,SFTN3 ;の最下位BIT
E402 CD 7A E6       502   CALL LDAHL
E405 4F             503   LD C,A ;
E406 23             504   INC HL
E407 CD 7A E6       505   CALL LDAHL
E40A E6 0F          506   AND $F
E40C 47             507   LD B,A ;BC=NEXT CRS
E40D 18 1D          508   JR SFTBC2
E40F                509  SFTN3 ;12BIT FAT用
E40F 23             510   INC HL
E410 CD 7A E6       511   CALL LDAHL
E413 E6 F0          512   AND $F0
E415 4F             513   LD C,A ;
E416 23             514   INC HL ;BC=NEXT CRS
E417 CD 7A E6       515   CALL LDAHL
E41A 47             516   LD B,A
E41B B7             517   OR A
E41C CB 18          518   RR B
E41E CB 19          519   RR C
E420 CB 18          520   RR B
E422 CB 19          521   RR C
E424 CB 18          522   RR B ;以上
E426 CB 19          523   RR C ;&B1111111111110000を
E428 CB 18          524   RR B ;&B0000111111111111に
E42A CB 19          525   RR C ;シフトしている
E42C                526  SFTBC2
E42C ED 43 95 E0    527   LD (#CRS),BC ;BC=NEXT CRS
E430 78             528   LD A,B
E431 B1             529   OR C
E432 28 15          530   JR Z,NCSRET0 ;FATが0?
E434 21 F7 0F       531   LD HL,$FF7 ;MS-DOSでは
E437 B7             532   OR A       ;$FF7以上で
E438 ED 42          533   SBC HL,BC  ;FATの終わり
E43A D0             534   RET NC     ;を示す
E43B 3A 97 E0       535   LD A,(#MAC)
E43E FE 02          536   CP 2 ;MS-DOS 2HD
E440 28 07          537   JR Z,NCSRET0
E442 79             538   LD A,C
E443 D6 FD          539   SUB $FD
E445 32 8F EB       540   LD (EDD),A
E448 C9             541   RET
E449                542  NCSRET0
E449 3E 01          543   LD A,1
E44B 32 8F EB       544   LD (EDD),A
E44E C9             545   RET
E44F                546
E44F                547  SFAT3 ;X1 2D
E44F 2A 8D E0       548   LD HL,(#FF)  ;FAT ADR
E452 ED 5B 95 E0    549   LD DE,(#CRS) ;クラスター
E456 19             550   ADD HL,DE ;たすだけ
E457 AF             551   XOR A
E458 32 96 E0       552   LD (#CRS+1),A
E45B CD 7A E6       553   CALL LDAHL
E45E 32 95 E0       554   LD (#CRS),A
E461 FE 80          555   CP $80 ;X1では$70から
E463 D8             556   RET C  ;$7FでFAT終了
E464 D6 7F          557   SUB $7F
E466 32 8F EB       558   LD (EDD),A
E469 C9             559   RET
E46A                560
E46A                561  ;未使用FATをさがす
E46A                562  ; & 終了CODEを書き込め
E46A                563  SEC00 ;IN ED,MAC,CRS OUT CRS
E46A 3A 97 E0       564   LD A,(#MAC)
E46D 01 7A 01       565    LD BC,$17A
E470 3D             566   DEC A
E471 28 0F          567   JR Z,SECSKSK
E473 01 C7 04       568    LD BC,1223
E476 3D             569   DEC A
E477 28 09          570   JR Z,SECSKSK
E479 01 50 00       571    LD BC,$50
E47C 3D             572   DEC A
E47D 28 03          573   JR Z,SECSKSK
E47F 01 63 01       574    LD BC,355
E482                575  SECSKSK ;BC=最終FAT
E482 3A 96 EB       576   LD A,(ED)
E485 B7             577   OR A
E486 28 22          578   JR Z,SECSK0 IF NOT 終了処理
E488 C6 7F          579    ADD A,$7F
E48A 26 00          580    LD H,0
E48C 6F             581    LD L,A
E48D 3A 97 E0       582    LD A,(#MAC)
E490 FE 01          583    CP 1
E492 28 11          584    JR Z,SECSK1
E494 FE 03          585    CP 3
E496 28 0D          586    JR Z,SECSK1
E498 21 FF 0F       587     LD HL,$FFF
E49B FE 02          588     CP 2
E49D 28 06          589     JR Z,SECSK1
E49F 3A 96 EB       590     LD A,(ED)
E4A2 C6 FD          591     ADD A,$FD
E4A4 6F             592     LD L,A
E4A5                593  SECSK1
E4A5 22 95 E0       594    LD (#CRS),HL ;HL=FATの
E4A8 B7             595    OR A         ;終了CODE
E4A9 C9             596    RET
E4AA                597  SECSK0 ;未使用さがし
E4AA 2A 95 E0       598   LD HL,(#CRS)
E4AD                599  SECLP0
E4AD 23             600    INC HL ;1個づつ調べる
```

▶なんて悲しいことだろう。今月の付録でいったい何人のX68000ユーザーが笑い死んだかと思うと。笑いが止まらんぜ，となな。　松永　直哉（21）東京都

```
E4AE 3A 97 E0      601     LD A,(#MAC)
E4B1 FE 01         602     CP 1
E4B3 20 05         603     JR NZ,SECMN1 ;IF X12HD以外
E4B5 7D            604      LD A,L       ;HLが$80
E4B6 FE 80         605      CP $80       ;以上に
E4B8 30 F3         606      JR NC,SECLP0 ;なるまで
E4BA               607   SECMN1
E4BA C5            608     PUSH BC
E4BB E5            609     PUSH HL
E4BC 22 95 E0      610     LD (#CRS),HL
E4BF B7            611      OR A
E4C0 ED 42         612      SBC HL,BC ;FATの最後か?
E4C2 30 12         613      JR NC,SECSK2 ;IF 最後
E4C4 CD AE E3      614     CALL SECFAT2  ;次は
E4C7 E1            615     POP HL
E4C8 C1            616     POP BC
E4C9 ED 5B 95 E0   617     LD DE,(#CRS) ;DE=FAT
E4CD 7A            618     LD A,D       ;
E4CE B3            619     OR E         ;
E4CF 20 DC         620     JR NZ,SECLP0 ;IF DE<>0
E4D1 22 95 E0      621     LD (#CRS),HL ;HL=未使用
E4D4 B7            622     OR A         ;クラスター
E4D5 C9            623     RET
E4D6               624   SECSK2 ;ディスクが一杯です
E4D6 E1            625    POP HL
E4D7 C1            626    POP BC
E4D8 C3 3C E0      627    JP #ERR5 ;エラー
E4DB               628
E4DB               629   ;クラスター番号を
E4DB               630   ;FATへ書き込む
E4DB               631   WTFAT2 ;IN CRSS,CRS,FF,MAC
E4DB 3A 97 E0      632    LD A,(#MAC)
E4DE 3D            633    DEC A
E4DF 28 09         634    JR Z,SFAT11
E4E1 3D            635    DEC A
E4E2 28 21         636    JR Z,SFAT22
E4E4 3D            637    DEC A
E4E5 CA 52 E5      638    JP Z,SFAT33
E4E8 18 1B         639    JR SFAT22
E4EA               640
E4EA               641   SFAT11 ;X12HD
E4EA 2A 8D E0      642    LD HL,(#FF)
E4ED ED 5B 90 EB   643    LD DE,(CRSS)
E4F1 19            644    ADD HL,DE
E4F2 EB            645    EX DE,HL ;DE=FAT
E4F3 3A 95 E0      646    LD A,(#CRS)
E4F6 CD 96 E6      647    CALL LDDEA
E4F9 EB            648    EX DE,HL  ;
E4FA 11 80 00      649    LD DE,128 ;
E4FD 19            650    ADD HL,DE ;
E4FE EB            651    EX DE,HL  ;DE=DE+128
E4FF 3A 96 E0      652    LD A,(#CRS+1)
E502 C3 96 E6      653    JP LDDEA
E505               654
E505               655   SFAT22 ;MS-DOS 2HD&2D
E505 2A 90 EB      656    LD HL,(CRSS)
E508 CB 3C         657    SRL H ;
E50A CB 1D         658    RR L  ;HL=HL/2
E50C 08            659    EX AF,AF' ;C FLUG
E50D E5            660    PUSH HL   ;
E50E 29            661    ADD HL,HL ;
E50F D1            662    POP DE    ;
E510 19            663    ADD HL,DE ;HL=(CRS)/2*3
E511 EB            664    EX DE,HL     ;
E512 2A 8D E0      665    LD HL,(#FF) ;FAT ADR
E515 19            666    ADD HL,DE    ;
E516 EB            667    EX DE,HL     ;HL=FAT ADR+
E517 08            668    EX AF,AF'
E518 38 14         669    JR C,SFTN33
E51A 3A 95 E0      670     LD A,(#CRS)
E51D CD 96 E6      671     CALL LDDEA
E520 13            672     INC DE
E521 CD 8A E6      673     CALL LDADE ;もう1つの
E524 E6 F0         674     AND $F0    ;FATと合わせ
E526 47            675     LD B,A     ;技
E527 3A 96 E0      676     LD A,(#CRS+1)
E52A B0            677     OR B       ;
E52B C3 96 E6      678     JP LDDEA   ;
E52E               679   SFTN33
E52E 3A 96 E0      680    LD A,(#CRS+1)
E531 47            681    LD B,A
E532 3A 95 E0      682    LD A,(#CRS)
E535 B7            683    OR A
E536 17            684    RLA
E537 CB 10         685    RL B
E539 17            686    RLA
E53A CB 10         687    RL B
E53C 17            688    RLA  ;以上
E53D CB 10         689    RL B ;&B0000111111111111を
E53F 17            690    RLA  ;&B1111111111110000に
E540 CB 10         691    RL B ;なおしている
E542 4F            692    LD C,A
E543 13            693    INC DE
E544 CD 8A E6      694    CALL LDADE
E547 E6 0F         695    AND $0F
E549 B1            696    OR C ;合わせ技
E54A CD 96 E6      697    CALL LDDEA
E54D 13            698    INC DE
E54E 78            699    LD A,B
E54F C3 96 E6      700    JP LDDEA
E552               701
E552               702   SFAT33 ;X1 2D
E552 2A 8D E0      703    LD HL,(#FF)
E555 ED 5B 90 EB   704    LD DE,(CRSS)
E559 19            705    ADD HL,DE
E55A EB            706    EX DE,HL
E55B 3A 95 E0      707    LD A,(#CRS)
E55E C3 96 E6      708    JP LDDEA
E561               709
E561               710   ;/ DELETE FAT /
E561               711   ;
E561               712   DELFAT ;IN ECRS
E561 2A 7C E0      713    LD HL,(#ECRS)
E564               714   DFATLP
E564 22 95 E0      715     LD (#CRS),HL
E567 22 90 EB      716     LD (CRSS),HL
E56A CD AE E3      717     CALL SECFAT2 ;位置
E56D 2A 95 E0      718     LD HL,(#CRS)
E570 E5            719     PUSH HL
E571 21 00 00      720      LD HL,0 ;0を書き込む
E574 22 95 E0      721      LD (#CRS),HL
E577 CD DB E4      722      CALL WTFAT2 ;WRITE
E57A E1            723     POP HL
E57B 3A 8F EB      724     LD A,(EDD) ;FATが
E57E B7            725     OR A        ;とぎれるまで
E57F 28 E3         726     JR Z,DFATLP
E581 C9            727    RET
E582               728
E582               729   ;( FORMAT別 DATA SET )
E582               730   ;
E582               731   VAR ;IN MAC
E582 C5            732    PUSH BC
E583 D5            733    PUSH DE
E584 E5            734    PUSH HL
E585 DD E5         735    PUSH IX
E587 3A 97 E0      736    LD A,(#MAC)
E58A DD 21 00 E1   737    LD IX,DVCDT1 ;X1 2HD
E58E 3D            738    DEC A
E58F 28 12         739    JR Z,VALUEJP
E591 DD 21 0D E1   740    LD IX,DVCDT2 ;MS-DOS2HD
E595 3D            741    DEC A
E596 28 0B         742    JR Z,VALUEJP
E598 DD 21 1A E1   743    LD IX,DVCDT3 ;X1 2D
E59C 3D            744    DEC A
E59D 28 04         745    JR Z,VALUEJP
E59F DD 21 27 E1   746    LD IX,DVCDT4 ;MS-DOS2D
E5A3               747   VALUEJP
E5A3 DD 6E 00      748     LD L,(IX+0)   ;セクター
E5A6 DD 66 01      749     LD H,(IX+1)   ;の長さ
E5A9 22 9F E0      750     LD (#SLNG),HL ;
E5AC DD 7E 02      751     LD A,(IX+2) ;1TRACK内の
E5AF 32 82 EB      752     LD (LSCT),A ;セクター数
E5B2 DD 7E 03      753     LD A,(IX+3) ;1クラスター内の
E5B5 32 95 EB      754     LD (RIC),A  ;レコード数
E5B8 DD 5E 04      755     LD E,(IX+4)   ;クラスターの
E5BB DD 56 05      756     LD D,(IX+5)   ;大きさ
E5BE ED 53 93 EB   757     LD (CSBYT),DE ;(バイト)
E5C2 2A 8F E0      758     LD HL,(#BSIZ) ;BUFFの大きさ
E5C5 CD 44 E6      759     CALL DIVHD  ;#BSIZ/CSBYT
E5C8 32 8A EB      760     LD (BSCS),A ;クラスター何個分
E5CB DD 6E 06      761     LD L,(IX+6) ;
E5CE DD 66 07      762     LD H,(IX+7) ;FAT開始位置
E5D1 22 83 EB      763     LD (RS),HL  ;(レコード)
E5D4 DD 6E 08      764     LD L,(IX+8) ;FAT終了位置
E5D7 DD 66 09      765     LD H,(IX+9) ;
E5DA 22 85 EB      766     LD (RE),HL  ;
E5DD DD 7E 0A      767     LD A,(IX+10)
E5E0 32 87 EB      768     LD (LTR),A  ;最終トラック
E5E3 DD 6E 0B      769     LD L,(IX+11)
E5E6 DD 66 0C      770     LD H,(IX+12)
E5E9 22 88 EB      771     LD (RRC),HL ;最終レコード
E5EC DD E1         772    POP IX
E5EE E1            773    POP HL
E5EF D1            774    POP DE
E5F0 C1            775    POP BC
E5F1 C9            776    RET
E5F2               777
E5F2               778   ;( FATをDISKへ書き込む )
E5F2               779   ;
E5F2               780   FATWT ;IN FF,SLG,RS,RE
E5F2 3E 01         781    LD A,1      ;
E5F4 32 A1 E0      782    LD (#FRWF),A ;WRITE
E5F7 CD 21 E6      783    CALL FATRW  ;
E5FA 3A 97 E0      784    LD A,(#MAC)
E5FD FE 01         785    CP 1
E5FF 28 1B         786    JR Z,FATWTN2
E601 FE 03         787    CP 3
E603 28 17         788    JR Z,FATWTN2
E605 21 03 00      789     LD HL,3 ;MS-DOSはFATを
E608 22 83 EB      790     LD (RS),HL
E60B 23            791     INC HL  ;2つ作る
E60C 22 85 EB      792     LD (RE),HL
E60F CD 21 E6      793     CALL FATRW
E612 21 01 00      794     LD HL,1
E615 22 83 EB      795     LD (RS),HL
E618 23            796     INC HL
E619 22 85 EB      797     LD (RE),HL
E61C               798   FATWTN2
E61C AF            799    XOR A
E61D 32 A1 E0      800    LD (#FRWF),A
E620 C9            801    RET
E621               802
E621               803   FATRW ;FAT読み書き
E621 D5            804    PUSH DE
E622 E5            805    PUSH HL
E623 2A 8D E0      806    LD HL,(#FF) ;FAT ADR
E626 22 91 E0      807    LD (#BF),HL ;BUFF
E629 2A 83 EB      808    LD HL,(RS)  ;開始REC
E62C 22 93 E0      809    LD (#REC),HL
E62F               810   FATRDLP
E62F CD 00 E0      811     CALL #RWREC
E632 38 0D         812     JR C,FLTFT
E634 2A 85 EB      813     LD HL,(RE)
E637 ED 5B 93 E0   814     LD DE,(#REC)
E63B B7            815     OR A
E63C ED 52         816     SBC HL,DE ;最終?
E63E 30 EF         817     JR NC,FATRDLP
E640 B7            818     OR A
E641               819   FLTFT
E641 E1            820    POP HL
E642 D1            821    POP DE
E643 C9            822    RET
E644               823
E644               824   ;ただの整数割り算
E644               825   DIVHD ;DE=HL¥DE,HL=HL MOD DE
E644 3E 10         826    LD A,16
E646 4B            827    LD C,E
E647 42            828    LD B,D
E648 EB            829    EX DE,HL
E649 21 00 00      830    LD HL,0
E64C               831   DIVLP1
E64C EB            832     EX DE,HL
E64D 29            833     ADD HL,HL
E64E EB            834     EX DE,HL
E64F ED 6A         835     ADC HL,HL
E651 CA 5D E6      836     JP Z,DIVI4
E654 ED 42         837     SBC HL,BC
E656 30 04         838     JR NC,DIVI3
E658 09            839     ADD HL,BC
E659 C3 5D E6      840     JP DIVI4
E65C               841   DIVI3
E65C 13            842     INC DE
E65D               843   DIVI4
E65D 3D            844     DEC A
E65E C2 4C E6      845     JP NZ,DIVLP1
E661 7B            846    LD A,E
E662 C9            847    RET
E663               848
E663               849   ;ただの整数掛け算
E663               850   MLTHD ;HL=HL*DE
E663 C5            851    PUSH BC
E664 D5            852    PUSH DE
E665 3E 10         853    LD A,16
E667 44            854    LD B,H
E668 4D            855    LD C,L
E669 21 00 00      856    LD HL,0
E66C               857   MLTLP
E66C 29            858     ADD HL,HL
E66D EB            859     EX DE,HL
E66E 29            860     ADD HL,HL
E66F EB            861     EX DE,HL
E670 D2 74 E6      862     JP NC,MULT2
E673 09            863      ADD HL,BC
E674               864   MULT2
E674 3D            865     DEC A
E675 20 F5         866     JR NZ,MLTLP
E677 D1            867    POP DE
E678 C1            868    POP BC
E679 C9            869    RET
E67A               870
E67A               871   ;以下はBUFFにアクセス
E67A               872   ;するための命令群
E67A               873   ;これによって
E67A               874   ;バンクやI/Oに関係なく
E67A               875   ;BUFFを設定できる
E67A               876   ;
E67A               877   LDAHL ;=LD A,(HL)
E67A CD B3 E6      878    CALL BGIOMM ;バンク替えなど
E67D 7E            879    LD A,(HL)
E67E D2 D9 E6      880    JP NC,BGRST ;もとへもどす
E681 C5            881    PUSH BC
E682 44            882    LD B,H
E683 4D            883    LD C,L
E684 ED 78         884    IN A,(C)
E686 C1            885    POP BC
E687 C3 D9 E6      886    JP BGRST
E68A               887   ;
E68A               888   LDADE ;=LD A,(DE)
E68A EB            889    EX DE,HL
E68B CD 7A E6      890    CALL LDAHL
E68E EB            891    EX DE,HL
E68F C9            892    RET
E690               893   ;
E690               894   LDHLA ;=LD (HL),A
E690 EB            895    EX DE,HL
E691 CD 96 E6      896    CALL LDDEA
E694 EB            897    EX DE,HL
E695 C9            898    RET
E696               899
E696               900   ;
E696               901   LDDEA ;=LD (DE),A
E696 08            902    EX AF,AF'
E697 CD B3 E6      903    CALL BGIOMM
E69A 38 05         904    JR C,LDDEASK
E69C 08            905    EX AF,AF'
E69D 12            906    LD (DE),A
E69E C3 D9 E6      907    JP BGRST
E6A1               908   LDDEASK
E6A1 C5            909    PUSH BC
E6A2 08            910    EX AF,AF'
E6A3 42            911    LD B,D
E6A4 4B            912    LD C,E
E6A5 ED 79         913    OUT (C),A
E6A7 C1            914    POP BC
E6A8 C3 D9 E6      915    JP BGRST
E6AB               916
E6AB               917   ;デバイスにより
E6AB               918   ;バンク切り替えなど
E6AB               919   ;
E6AB               920   BGDN
E6AB C5            921    PUSH BC
E6AC 3A 8B E0      922    LD A,(#DN)
E6AF D6 04         923    SUB 4
E6B1 18 06         924    JR BGIOMM2
E6B3               925
E6B3               926   ;BUFFにより
E6B3               927   ;バンク切り替えなど
E6B3               928   ;
E6B3               929   BGIOMM
E6B3 C5            930    PUSH BC
E6B4 3A 9A E0      931    LD A,(#IOMM)
E6B7 D6 01         932    SUB 1
E6B9               933   BGIOMM2
E6B9 3F            934    CCF
E6BA DC BF E6      935    CALL C,BGBANK
E6BD C1            936    POP BC
E6BE C9            937    RET
E6BF               938
E6BF               939   BGBANK
E6BF D6 01         940    SUB 1 ;
E6C1 D8            941    RET C ;IF MEM0
E6C2 20 0C         942    JR NZ,BGBANK2
E6C4 01 D0 1F      943     LD BC,$1FD0 ;MEM1
E6C7 3A A2 E0      944     LD A,(#WFD0)
E6CA F6 10         945     OR $10
E6CC ED 79         946     OUT (C),A
E6CE 37            947     SCF ;GRAMならCARRY
E6CF C9            948     RET
E6D0               949   BGBANK2 ;バンク
E6D0 F3            950    DI
E6D1 3D            951    DEC A
E6D2 01 00 0B      952    LD BC,$B00
E6D5 ED 79         953    OUT (C),A
E6D7 B7            954    OR A
E6D8 C9            955    RET
E6D9               956
E6D9               957   ;全部もとへ戻す
E6D9               958   BGRST
E6D9 08            959    EX AF,AF'
E6DA C5            960    PUSH BC
E6DB 01 D0 1F      961    LD BC,$1FD0
E6DE 3A A2 E0      962    LD A,(#WFD0)
E6E1 ED 79         963    OUT (C),A ;GRAM
E6E3 01 00 0B      964    LD BC,$B00
E6E6 3E 10         965    LD A,$10
E6E8 ED 79         966    OUT (C),A ;バンク
E6EA C1            967    POP BC
E6EB 08            968    EX AF,AF'
E6EC FB            969    EI
E6ED C9            970    RET
E6EE               971
E6EE               972   ;HL=HL+(#DN)
E6EE               973   DNST
E6EE 06 00         974    LD B,0
E6F0 3A 8B E0      975    LD A,(#DN)
E6F3 4F            976    LD C,A
E6F4 09            977    ADD HL,BC
E6F5 C9            978    RET
E6F6               979
E6F6               980   ;/ 直接I/Oをいじったり
E6F6               981   ;  FDCをハンドルする部分 /
E6F6               982   SETLD
E6F6 01 FF 0F      983    LD BC,$FFF
E6F9 ED 48         984    IN C,(C)
E6FB C9            985    RET
E6FC               986
E6FC               987   ;
E6FC               988   MTRON
E6FC 01 FE 0F      989    LD BC,$FFE
E6FF 3A 97 E0      990    LD A,(#MAC)
E702 FE 03         991    CP 3
E704 DA 08 E7      992    JP C,MTRHD ;IF 2HD
E707 0C            993     INC C ;BC=$FFF
E708               994   MTRHD
E708 ED 78         995    IN A,(C) ;SET HD OR LD
E70A 3A 7F EB      996    LD A,(SID) ;DISKの表裏
E70D 87            997    ADD A,A    ;&B00001から
E70E 87            998    ADD A,A    ;
E70F 87            999    ADD A,A    ;
E710 87           1000    ADD A,A    ;&B10000へ
E711 C6 80        1001    ADD A,$80  ;MOTOR ON
E713 4F           1002    LD C,A     ;C=C+
E714 3A 8B E0     1003    LD A,(#DN) ;DRIVE番号
E717 81           1004    ADD A,C    ;
E718 0E FC        1005    LD C,$FC ;BC=$FFC
E71A ED 79        1006    OUT (C),A
E71C C3 08 EA     1007    JP WTMTR ;WAIT
E71F              1008
E71F              1009   ;
E71F              1010   MTROF
E71F 01 FC 0F     1011    LD BC,$FFC
E722 3A 8B E0     1012    LD A,(#DN)
E725 ED 79        1013    OUT (C),A
E727 C9           1014    RET
E728              1015
E728              1016   ;ヘッドを0TRACKへ
E728              1017   ;
E728              1018   RSTOR
E728 C5           1019    PUSH BC
E729 E5           1020    PUSH HL
E72A CD FC E6     1021    CALL MTRON ;MOTOR ON
E72D 38 1B        1022    JR C,RSTRT ;エラー
E72F 01 F8 0F     1023    LD BC,$FF8 ;
E732 3A A6 E0     1024    LD A,(#RSCMD) ;COMMAND
E735 ED 79        1025    OUT (C),A
E737 CD 0D EA     1026    CALL WTTP1 ;WAIT
E73A 38 0E        1027    JR C,RSTRT
E73C 21 DA EC     1028    LD HL,&TR4 ;
E73F CD EE E6     1029    CALL DNST  ;
E742 AF           1030    XOR A      ;
E743 77           1031    LD (HL),A  ;TRACK WORK
E744 32 80 EB     1032    LD (TRN),A ;を0にする
E747 CD 1F E7     1033    CALL MTROF
E74A              1034   RSTRT
E74A E1           1035    POP HL
E74B C1           1036    POP BC
E74C C9           1037    RET
E74D              1038
E74D              1039   ;TRACK番号をCHECK
E74D              1040   ;しないSEEK
E74D              1041   SEEKT
```

▶ Oh！X を買った。中間テスト中にだ。Yet Another Colum にハマッた。勉強どころではなくなった。
安保　和幸 (16) 愛知県

```
E74D 21 DA EC     1042   LD HL,&TR4
E750 CD EE E6     1043   CALL DNST
E753 01 F9 0F     1044   LD BC,$FF9
E756 7E           1045   LD A,(HL) ;現在のTRACK
E757 ED 79        1046   OUT (C),A ;
E759 3A A7 E0     1047   LD A,(#SKCMD1) ;COMMAND
E75C 57           1048   LD D,A
E75D 3A 80 EB     1049   LD A,(TRN)
E760 CD 9A E7     1050   CALL TRLPS ;実行
E763 D8           1051   RET C
E764 21 DA EC     1052   LD HL,&TR4
E767 CD EE E6     1053   CALL DNST
E76A 3A 80 EB     1054   LD A,(TRN)
E76D 77           1055   LD (HL),A ;WORK更新
E76E C9           1056   RET
E76F              1057
E76F              1058  SEEK ;通常のSEEK
E76F 21 DA EC     1059    LD HL,&TR4
E772 06 00        1060    LD B,0
E774 3A 8B E0     1061    LD A,(#DN)
E777 4F           1062    LD C,A
E778 09           1063    ADD HL,BC
E779 01 F9 0F     1064   LD BC,$FF9
E77C 7E           1065   LD A,(HL) ;A=TRACK
E77D ED 79        1066   OUT (C),A
E77F 3A 80 EB     1067   LD A,(TRN)
E782 BE           1068   CP (HL)
E783 C8           1069   RET Z ;IF 同一TACK
E784 3A A8 E0     1070    LD A,(#SKCMD2) ;COMMAND
E787 57           1071    LD D,A
E788 3A 80 EB     1072    LD A,(TRN)
E78B CD 9A E7     1073    CALL TRLPS ;実行
E78E D8           1074    RET C
E78F 21 DA EC     1075   LD HL,&TR4
E792 CD EE E6     1076   CALL DNST
E795 3A 80 EB     1077   LD A,(TRN)
E798 77           1078   LD (HL),A ;WORK更新
E799 C9           1079   RET
E79A              1080
E79A              1081  TRLPS ;A=TRACK D=COMMAND
E79A 01 FB 0F     1082   LD BC,$FFB
E79D ED 79        1083   OUT (C),A
E79F 0E F8        1084   LD C,$F8
E7A1 ED 51        1085   OUT (C),D
E7A3 C3 0D EA     1086   JP WTTP1 WAIT
E7A6              1087
E7A6              1088  ;( 1レコード入出力 )
E7A6              1089  ; DMA を使用
E7A6              1090  RWREC ;IN BF,SLNG,REC,LSCT,DN
                        OUT BF,REC
E7A6 3A 8B E0     1091   LD A,(#DN)
E7A9 FE 04        1092   CP 4
E7AB D2 C5 E8     1093   JP NC,RAMRC    ;RAM-DISK
E7AE              1094
E7AE 3A A1 E0     1095   LD A,(#FRWF)   ;FLOPPY-DISK
E7B1 B7           1096   OR A
E7B2 20 14        1097   JR NZ,WTREC ;WRITE
E7B4 2A 9F E0     1098   LD HL,(#SLNG)
E7B7 2B           1099   DEC HL
E7B8 22 0E EB     1100   LD (SLNG1),HL ;セクター長
E7BB 2A 91 E0     1101   LD HL,(#BF)
E7BE 22 14 EB     1102   LD (BUFF1),HL ;BUFF ADR
E7C1 21 09 EB     1103   LD HL,RDDMAD ;DMA DATA
E7C4 3E 80        1104   LD A,$80 ;READ COMMAND
E7C6 18 12        1105   JR SKPREC
E7C8              1106
E7C8              1107  WTREC ;WRITE REC
E7C8 2A 9F E0     1108   LD HL,(#SLNG)
E7CB 2B           1109   DEC HL
E7CC 22 1E EB     1110   LD (SLNG2),HL
E7CF 2A 91 E0     1111   LD HL,(#BF)
E7D2 22 1C EB     1112   LD (BUFF2),HL
E7D5 21 19 EB     1113   LD HL,WTDMAD ;DMA DATA
E7D8 3E A0        1114   LD A,$A0 ;WRITE COMMAND
E7DA              1115  SKPREC ;読み書き共通
E7DA 22 8D EB     1116   LD (DMACMD),HL
E7DD 32 8C EB     1117   LD (FDCCMD),A
E7E0 C5           1118   PUSH BC
E7E1 D5           1119   PUSH DE
E7E2 CD A9 E8     1120   CALL GMCHG ;バンク切り替えなど
E7E5 CD 78 E8     1121   CALL RECTRN ;DATA SET
E7E8 D4 FC E6     1122   CALL NC,MTRON
E7EB D4 6F E7     1123   CALL NC,SEEK
E7EE 38 3C        1124   JR C,SSTPM
E7F0 01 FA 0F     1125   LD BC,$FFA
E7F3 3A 81 EB     1126   LD A,(SCT)
E7F6 ED 79        1127   OUT (C),A ;セクター
E7F8 0E F8        1128   LD C,$F8
E7FA 3A 8C EB     1129   LD A,(FDCCMD)
E7FD ED 79        1130   OUT (C),A ;FDC COMMAND
E7FF 2A 8D EB     1131   LD HL,(DMACMD)
E802 CD 88 EA     1132   CALL SETDMA
E805 CD 12 EA     1133   CALL WTTP22 ;WAIT
E808 38 22        1134   JR C,SSTPM
E80A 3E 83        1135    LD A,$83
E80C 01 80 1F     1136    LD BC,$1F80
E80F ED 79        1137    OUT (C),A ;RESTDMA
E811 2A 91 E0     1138   LD HL,(#BF)    ;BUFF ADR
E814 ED 5B 9F E0  1139   LD DE,(#SLNG) ;セクター長
E818 19           1140   ADD HL,DE      ;BF=BF+SLNG
E819 22 91 E0     1141   LD (#BF),HL    ;
E81C 2A 93 E0     1142   LD HL,(#REC) ;レコード番号
E81F 23           1143   INC HL         ;REC=REC+1
E820 22 93 E0     1144   LD (#REC),HL ;
E823 01 FC 0F     1145    LD BC,$FFC
E826 3A 8B E0     1146    LD A,(#DN)
E829 ED 79        1147    OUT (C),A ;MTROF
E82B B7           1148    OR A
E82C              1149  SSTPM
E82C 01 D0 1F     1150    LD BC,$1FD0   ;
E82F 3A A2 E0     1151    LD A,(#WFD0)  ;
E832 ED 79        1152    OUT (C),A     ;
E834 01 00 0B     1153    LD BC,$B00    ;高速化の
E837 3E 10        1154    LD A,$10      ;ため独立
E839 ED 79        1155    OUT (C),A     ;している
E83B FB           1156    EI            ;=BGRST
E83C D1           1157   POP DE
E83D C1           1158   POP BC
E83E C9           1159   RET
E83F              1160
E83F              1161  ;WRITE TRACK(FORMAT専用)
E83F              1162  WTTRC ;構造は同じ
E83F C5           1163   PUSH BC
E840 D5           1164   PUSH DE
E841 2A 9F E0     1165   LD HL,(#SLNG)
E844 2B           1166   DEC HL
E845 22 1E EB     1167   LD (SLNG2),HL
E848 2A 91 E0     1168   LD HL,(#BF)
E84B 22 1C EB     1169   LD (BUFF2),HL
E84E CD A9 E8     1170   CALL GMCHG
E851 CD 78 E8     1171   CALL RECTRN
E854 D4 FC E6     1172   CALL NC,MTRON
E857 D4 4D E7     1173   CALL NC,SEEKT
E85A 38 16        1174   JR C,SSTPM2
E85C 21 19 EB     1175   LD HL,WTDMAD
E85F CD 88 EA     1176   CALL SETDMA
E862 01 F8 0F     1177   LD BC,$FF8
E865 3E F0        1178   LD A,$F0  ;WRITE TRACK
E867 ED 79        1179   OUT (C),A ;FDC COMMAND
E869 CD 12 EA     1180   CALL WTTP22
E86C D4 80 EA     1181   CALL NC,RESTDMA
E86F D4 1F E7     1182   CALL NC,MTROF
E872              1183  SSTPM2
E872 CD D9 E6     1184   CALL BGRST
E875 D1           1185   POP DE
E876 C1           1186   POP BC
E877 C9           1187   RET
E878              1188
E878              1189  ;レコードから
E878              1190  ;トラック、セクター
E878              1191  ;サイドを計算する
E878              1192  RECTRN
E878 ED 5B 93 E0  1193   LD DE,(#REC)
E87C 2A 88 EB     1194   LD HL,(RRC)
E87F B7           1195   OR A
E880 ED 52        1196   SBC HL,DE
E882 DA 79 EA     1197   JP C,ERR9 ;OVER REC
E885 EB           1198   EX DE,HL ;HL=レコード
E886 3A 82 EB     1199   LD A,(LSCT) ;セクター数*2
E889 87           1200   ADD A,A      ;で割る
E88A 5F           1201   LD E,A       ;(表裏)
E88B 16 00        1202   LD D,0       ;
E88D CD 44 E6     1203   CALL DIVHD   ;
E890 32 80 EB     1204   LD (TRN),A ;トラック
E893 1E 01        1205   LD E,1 ;裏
E895 3A 82 EB     1206   LD A,(LSCT)
E898 67           1207   LD H,A
E899 7D           1208   LD A,L ;L=余り
E89A 94           1209   SUB H  ;余り>LSCT?
E89B 30 02        1210   JR NC,SIDSK
E89D 84           1211    ADD A,H
E89E 1D           1212    DEC E ;表
E89F              1213  SIDSK
E89F 3C           1214   INC A
E8A0 32 81 EB     1215   LD (SCT),A ;セクター
E8A3 7B           1216   LD A,E
E8A4 32 7F EB     1217   LD (SID),A ;サイド
E8A7 B7           1218   OR A
E8A8 C9           1219   RET
E8A9              1220
E8A9              1221  GMCHG
E8A9 F3           1222   DI
E8AA CD B3 E6     1223   CALL BGIOMM
E8AD 30 0B        1224   JR NC,GMCHGSK
E8AF 3E 18        1225   LD A,$18 ;DMA DATA
E8B1 32 11 EB     1226   LD (IOMEM1),A
E8B4 3E 1C        1227   LD A,$1C ;
E8B6 32 20 EB     1228   LD (IOMEM2),A
E8B9 C9           1229   RET
E8BA              1230  GMCHGSK
E8BA 3E 10        1231   LD A,$10 ;DMA DATA
E8BC 32 11 EB     1232   LD (IOMEM1),A
E8BF 3E 14        1233   LD A,$14 ;
E8C1 32 20 EB     1234   LD (IOMEM2),A
E8C4 C9           1235   RET
E8C5              1236
E8C5              1237  ;RAM DISK アクセス
E8C5              1238  ;
E8C5              1239  RAMRC ;
E8C5 C5           1240   PUSH BC
E8C6 D5           1241   PUSH DE
E8C7 E5           1242   PUSH HL
E8C8 ED 5B 93 E0  1243   LD DE,(#REC)
E8CC 2A 88 EB     1244   LD HL,(RRC)
E8CF B7           1245   OR A
E8D0 ED 52        1246   SBC HL,DE
E8D2 DA 79 EA     1247   JP C,ERR9 ;OVER REC
E8D5 3A 8B E0     1248    LD A,(#DN)
E8D8 D6 16        1249    SUB 22
E8DA D2 9E E9     1250    JP NC,EMMRW ;EMMドライブ
E8DD              1251  ;MEM & バンク
E8DD 21 00 40     1252   LD HL,$4000  ;オフセット
E8E0 22 97 EB     1253   LD (SETOFF),HL
E8E3 21 2B EB     1254   LD HL,KENKO1
E8E6 11 06 00     1255   LD DE,6
E8E9 0E 00        1256   LD C,0
E8EB CD AB E6     1257   CALL BGDN ;バンク切り替え
E8EE 38 0B        1258   JR C,PATCHSK1
E8F0 E5           1259    PUSH HL
E8F1 21 00 00     1260    LD HL,0 ;オフセット
E8F4 22 97 EB     1261    LD (SETOFF),HL
E8F7 E1           1262    POP HL
E8F8 0C           1263    INC C      ;C+1
E8F9 19           1264    ADD HL,DE ;KENKO1+6
E8FA 19           1265    ADD HL,DE ;KENKO1+12
E8FB              1266  PATCHSK1
E8FB CD B3 E6     1267   CALL BGIOMM ;バンク切り替え
E8FE 38 02        1268   JR C,PATCHSK2
E900 0C           1269    INC C      ;C+2
E901 19           1270    ADD HL,DE ;KENKO1+18
E902              1271  PATCHSK2
E902              1272  ;        C=0   C=1  C=2
E902              1273  ;DRIVE   B     B G   G
E902              1274  ;BUFF    B(M)  G B   G
E902              1275  ;B=バンク M=メモリ G=GRAM
E902              1276  ;
E902 3A 9A E0     1277   LD A,(#IOMM)   ;
E905 3D           1278   DEC A          ;
E906 47           1279   LD B,A         ;
E907 3A 8B E0     1280   LD A,(#DN)     ;
E90A D6 04        1281   SUB 4          ;
E90C B8           1282   CP B           ;
E90D 28 0A        1283   JR Z,PATCHSK3 ;同一バンク
E90F 11 43 EB     1284   LD DE,KENKO5
E912 0D           1285   DEC C
E913 28 04        1286   JR Z,PATCHSK3
E915 CD D9 E6     1287    CALL BGRST ;裏バンク
E918 EB           1288    EX DE,HL
E919              1289  PATCHSK3
E919 11 1E 00     1290   LD DE,30
E91C 3A A1 E0     1291   LD A,(#FRWF)
E91F B7           1292   OR A
E920 28 01        1293   JR Z,PATCHSK4
E922 19           1294    ADD HL,DE ;WRITE
E923              1295  PATCHSK4
E923 11 51 E9     1296   LD DE,RMPATH1
E926 01 06 00     1297   LD BC,6
E929 ED B0        1298   LDIR ;自己書き換え
E92B CD 36 E9     1299    CALL RAMRWS
E92E CD D9 E6     1300   CALL BGRST
E931              1301  EMMRT
E931 B7           1302   OR A
E932 E1           1303   POP HL
E933 D1           1304   POP DE
E934 C1           1305   POP BC
E935 C9           1306   RET
E936              1307
E936              1308  ;R:BC->BC' W:BC'->BC
E936              1309  RAMRWS
E936 2A 93 E0     1310   LD HL,(#REC)
E939 ED 5B 9F E0  1311   LD DE,(#SLNG)
E93D CD 63 E6     1312   CALL MLTHD ;REC*SLNG
E940 ED 5B 97 EB  1313   LD DE,(SETOFF)
E944 19           1314   ADD HL,DE
E945 44           1315   LD B,H
E946 4D           1316   LD C,L ;BC=OF+REC*SLNG
E947 D9           1317   EXX
E948 ED 4B 91 E0  1318   LD BC,(#BF)
E94C D9           1319   EXX
E94D ED 5B 9F E0  1320   LD DE,(#SLNG)
E951              1321  RMRCLP
E951              1322  RMPATH1 ;CALL RMRW
E951              1323    DS 6 ;書き換え場所
E957 03           1324    INC BC
E958 D9           1325    EXX
E959 03           1326    INC BC
E95A D9           1327    EXX
E95B 1B           1328    DEC DE
E95C 7A           1329    LD A,D
E95D B3           1330    OR E
E95E 20 F1        1331    JR NZ,RMRCLP
E960 D9           1332   EXX
E961 ED 43 91 E0  1333   LD (#BF),BC
E965 D9           1334   EXX
E966 2A 93 E0     1335   LD HL,(#REC)
E969 23           1336   INC HL
E96A 22 93 E0     1337   LD (#REC),HL
E96D C9           1338   RET
E96E              1339
E96E              1340  ;裏バンクからREAD
E96E              1341  BFINS
E96E CD B3 E6     1342   CALL BGIOMM
E971 D9           1343   EXX
E972 CD 7A E9     1344   CALL DVINS2
E975 D9           1345   EXX
E976 C9           1346   RET
E977              1347  DVINS
E977 CD AB E6     1348   CALL BGDN
E97A              1349  DVINS2
E97A 0A           1350   LD A,(BC)
E97B D2 D9 E6     1351   JP NC,BGRST ;IF NOT MEM
E97E ED 78        1352   IN A,(C)
E980 C3 D9 E6     1353   JP BGRST
E983              1354
E983              1355  ;裏バンクへWRITE
E983              1356  BFOUS
E983 08           1357   EX AF,AF'
E984 CD B3 E6     1358   CALL BGIOMM
E987 D9           1359   EXX
E988 CD 91 E9     1360   CALL DVOUS2
E98B D9           1361   EXX
E98C C9           1362   RET
E98D              1363  DVOUS
E98D 08           1364   EX AF,AF'
E98E CD AB E6     1365   CALL BGDN
E991              1366  DVOUS2
E991 38 05        1367   JR C,DVOUS3 ;IF MEM
E993 08           1368   EX AF,AF'
E994 02           1369   LD (BC),A
E995 C3 D9 E6     1370   JP BGRST
E998              1371  DVOUS3
E998 08           1372   EX AF,AF'
E999 ED 79        1373   OUT (C),A
E99B C3 D9 E6     1374   JP BGRST
E99E              1375
E99E              1376  ;EMM 入出力
E99E              1377  ;
E99E              1378  EMMRW
E99E 87           1379   ADD A,A ;
E99F 87           1380   ADD A,A ;
E9A0 4F           1381   LD C,A  ;EMM?
E9A1 06 0D        1382   LD B,$D
E9A3 2A 93 E0     1383   LD HL,(#REC)
E9A6 ED 5B 9F E0  1384   LD DE,(#SLNG)
E9AA CD 63 E6     1385   CALL MLTHD
E9AD ED 69        1386   OUT (C),L ;下位
E9AF 03           1387   INC BC
E9B0 ED 61        1388   OUT (C),H ;中位
E9B2 03           1389   INC BC
E9B3 C5           1390   PUSH BC
E9B4 2A 93 E0     1391   LD HL,(#REC)
E9B7 11 00 01     1392   LD DE,256
E9BA CD 44 E6     1393   CALL DIVHD
E9BD C1           1394   POP BC
E9BE ED 79        1395   OUT (C),A ;上位 ADR
E9C0 03           1396   INC BC
E9C1 C5           1397   PUSH BC
E9C2 D9           1398   EXX
E9C3 C1           1399   POP BC
E9C4 D9           1400   EXX
E9C5 21 67 EB     1401    LD HL,REMMDT
E9C8 3A A1 E0     1402    LD A,(#FRWF)
E9CB B7           1403    OR A
E9CC 28 04        1404    JR Z,EMMPTH1 ;IF READ
E9CE 11 0C 00     1405     LD DE,12
E9D1 19           1406     ADD HL,DE ;REMMDT+12
E9D2              1407  EMMPTH1
E9D2 CD B3 E6     1408    CALL BGIOMM
E9D5 38 04        1409    JR C,EMMPTH2 ;IF GRAM
E9D7 11 06 00     1410     LD DE,6
E9DA 19           1411     ADD HL,DE ;REMMDT+6,18
E9DB              1412  EMMPTH2
E9DB 11 EB E9     1413   LD DE,EMMPTH
E9DE 01 06 00     1414   LD BC,6
E9E1 ED B0        1415   LDIR ;自己書き換え
E9E3 ED 4B 91 E0  1416   LD BC,(#BF)    ;BUFF ADR
E9E7 ED 5B 9F E0  1417   LD DE,(#SLNG) ;セクター長
E9EB              1418  EMMCLP
E9EB              1419  EMMPTH
E9EB              1420    DS 6 ;書き換え場所
E9F1 03           1421    INC BC
E9F2 1B           1422    DEC DE
E9F3 7A           1423    LD A,D
E9F4 B3           1424    OR E
E9F5 20 F4        1425    JR NZ,EMMCLP
E9F7 ED 43 91 E0  1426   LD (#BF),BC
E9FB 2A 93 E0     1427   LD HL,(#REC)
E9FE 23           1428   INC HL
E9FF 22 93 E0     1429   LD (#REC),HL
EA02 CD D9 E6     1430   CALL BGRST
EA05 C3 31 E9     1431   JP EMMRT
EA08              1432
EA08              1433  ;( WAIT )
EA08              1434  ;
EA08              1435  WTMTR ;MOTOR用
EA08 16 81        1436   LD D,$81
EA0A C3 14 EA     1437   JP WNBSY
EA0D              1438  WTTP1 ;READその他
EA0D 16 99        1439   LD D,$99
EA0F C3 14 EA     1440   JP WNBSY
EA12              1441  WTTP22 ;WRITE用
EA12 16 FD        1442   LD D,$FD
EA14              1443  WNBSY
EA14 01 F8 0F     1444   LD BC,$FF8
EA17 3A AC E0     1445   LD A,(#WAIT)
EA1A 5F           1446   LD E,A
EA1B              1447  WAITLP2
EA1B 21 00 10     1448    LD HL,$1000 ;*WAIT回
EA1E              1449  WNBSYLP
EA1E 2B           1450     DEC HL
EA1F 7C           1451     LD A,H
EA20 B5           1452     OR L
EA21 C2 29 EA     1453     JP NZ,WNBLP2
EA24 1D           1454     DEC E
EA25 20 F4        1455     JR NZ,WAITLP2
EA27 18 08        1456     JR DSTOP ;エラー
EA29              1457  WNBLP2
EA29 ED 78        1458    IN A,(C)
EA2B A2           1459    AND D
EA2C C2 1E EA     1460    JP NZ,WNBSYLP
EA2F B7           1461   OR A
EA30 C9           1462   RET
EA31              1463  DSTOP ;エラー
EA31 ED 78        1464   IN A,(C)
EA33 A2           1465   AND D
EA34 4F           1466   LD C,A
EA35 E6 40        1467   AND $40
EA37 C4 75 EA     1468   CALL NZ,ERR7
EA3A 20 09        1469   JR NZ,ERR72
EA3C 79           1470   LD A,C
EA3D E6 81        1471   AND $81
EA3F C4 5D EA     1472   CALL NZ,ERR1
EA42 CC 61 EA     1473   CALL Z,ERR2
EA45              1474  ERR72
EA45 3A 8B E0     1475   LD A,(#DN)
EA48 01 FC 0F     1476   LD BC,$FFC
EA4B ED 79        1477   OUT (C),A ;MOTOR OFF
EA4D CD 80 EA     1478   CALL RESTDMA ;RESET DMA
EA50 21 00 00     1479   LD HL,0
EA53 22 95 E0     1480   LD (#CRS),HL
EA56 37           1481   SCF ;エラーはCARRY FLUG!
EA57 C9           1482   RET
```

▶難関大学突破プログラム“浪人.X”を祝氏に作っていただきタイヤキ。

田中　優一（18）熊本県

```
EA58                1483  ;
EA58                1484  ERR0
EA58 AF             1485   XOR A
EA59 32 8C E0       1486   LD (#STOP),A
EA5C C9             1487   RET
EA5D                1488  ERR1
EA5D 3E 01          1489   LD A,1
EA5F 18 1A          1490   JR ERR
EA61                1491  ERR2
EA61 3E 02          1492   LD A,2
EA63 18 16          1493   JR ERR
EA65                1494  ERR3
EA65 3E 03          1495   LD A,3
EA67 18 12          1496   JR ERR
EA69                1497  ERR4
EA69 3E 04          1498   LD A,4
EA6B 18 0E          1499   JR ERR
EA6D                1500  ERR5
EA6D 3E 05          1501   LD A,5
EA6F 18 0A          1502   JR ERR
EA71                1503  ERR6
EA71 3E 06          1504   LD A,6
EA73 18 06          1505   JR ERR
EA75                1506  ERR7
EA75 3E 07          1507   LD A,7
EA77 18 02          1508   JR ERR
EA79                1509  ERR9
EA79 3E 09          1510   LD A,9
EA7B                1511  ERR
EA7B 32 8C E0       1512   LD (#STOP),A
EA7E 37             1513   SCF
EA7F C9             1514   RET
EA80                1515
EA80                1516  ;RESET DMA
EA80                1517  RESTDMA
EA80 3E 83          1518   LD A,$83
EA82 01 80 1F       1519   LD BC,$1F80
EA85 ED 79          1520   OUT (C),A
EA87 C9             1521   RET
EA88                1522
EA88                1523  ;DMAにCOMMANDをSET
EA88                1524  SETDMA
EA88 01 80 1F       1525   LD BC,$1F80
EA8B 5E             1526   LD E,(HL) ;(HL+1)
EA8C 23             1527   INC HL ;から(HL)回
EA8D                1528  STDMALP
EA8D 04             1529    INC B
EA8E ED A3          1530    OUTI
EA90 1D             1531    DEC E
EA91 20 FA          1532    JR NZ,STDMALP
EA93 C9             1533   RET
EA94                1534
EA94                1535  ;文字列中、改行
EA94                1536  ;まで切り出す
EA94                1537  ;独立ルーチン
EA94                1538  ;
EA94                1539  GYOKI
EA94 01 00 FE       1540   LD BC,$FE00
EA97 ED 5B 9D E0    1541   LD DE,(#MSBT)
EA9B 2A 91 E0       1542   LD HL,(#BF)
EA9E DD 21 02 EE    1543   LD IX,P256+2
EAA2                1544  GYOKILP
EAA2 7A             1545   LD A,D ;文字列終わり?
EAA3 B3             1546   OR E   ;
EAA4 20 0C          1547   JR NZ,GYOKISK ;IF NOT END
EAA6 3A E7 EC       1548    LD A,(&EOF2)
EAA9 DD BE FF       1549    CP (IX+$FF)
EAAC 20 01          1550    JR NZ,GYOEOFSK
EAAE 0D             1551     DEC C ;EOFを抜く
EAAF                1552  GYOEOFSK
EAAF AF             1553    XOR A
EAB0 18 17          1554    JR GYOKISK2
EAB2                1555  GYOKISK
EAB2 1B             1556   DEC DE
EAB3 CD 7A E6       1557   CALL LDAHL
EAB6 23             1558   INC HL
EAB7 FE 0D          1559   CP $D
EAB9 20 04          1560   JR NZ,GYONEND
EABB 3E 02          1561    LD A,2
EABD 18 0A          1562    JR GYOKISK2
EABF                1563  GYONEND
EABF DD 77 00       1564   LD (IX+0),A ;(IX)<-(HL)
EAC2 DD 23          1565   INC IX
EAC4 0C             1566   INC C
EAC5 10 DB          1567   DJNZ GYOKILP
EAC7 3E 01          1568   LD A,1
EAC9                1569  GYOKISK2
EAC9 32 00 EE       1570   LD (P256),A ;改行 FLUG
EACC 79             1571   LD A,C ;文字数
EACD 32 01 EE       1572   LD (P256+1),A
EAD0 22 91 E0       1573   LD (#BF),HL
EAD3 ED 53 9D E0    1574   LD (#MSBT),DE
EAD7 C9             1575   RET
EAD8                1576
EAD8                1577  ;フルパス作成
EAD8                1578  ;
EAD8                1579  SPATH ;IN (DE),B
EAD8 2A A9 E0       1580   LD HL,(#BPATH)
EADB DD 21 81 EE    1581   LD IX,P256+$81
EADF AF             1582   XOR A ;文字列長
EAE0 32 80 EE       1583   LD (P256+$80),A
EAE3 4F             1584   LD C,A
EAE4                1585  SPATHLP
EAE4 7E             1586   LD A,(HL)
EAE5 23             1587   INC HL
EAE6 B7             1588   OR A
EAE7 CA 65 EA       1589   JP Z,ERR3 ;END OF PATH
EAEA FE 3B          1590   CP ';' ;PATHの区切り
EAEC 28 08          1591   JR Z,SPATHSK
EAEE DD 77 00       1592   LD (IX+0),A ;下位DIR
EAF1 DD 23          1593   INC IX
EAF3 0C             1594   INC C
EAF4 18 EE          1595   JR SPATHLP
EAF6                1596  SPATHSK
EAF6 1A             1597   LD A,(DE)
EAF7 13             1598   INC DE
EAF8 DD 77 00       1599   LD (IX+0),A ;NAME
EAFB DD 23          1600   INC IX
EAFD 0C             1601   INC C
EAFE 10 F6          1602   DJNZ SPATHSK
EB00 79             1603   LD A,C ;文字列長
EB01 32 80 EE       1604   LD (P256+$80),A
EB04 22 A9 E0       1605   LD (#BPATH),HL
EB07 B7             1606   OR A
EB08 C9             1607   RET
EB09                1608
EB09                1609  ;/ DMA DATA /
EB09                1610  ;
EB09                1611  RDDMAD ;READ
EB09 0F 83 7D FB    1612   DB 15,$83,$7D,$FB,$0F
EB0D 0F
EB0E                1613  SLNG1
EB0E FF 00          1614    DW 256-1
EB10 2C             1615    DB $2C
EB11                1616  IOMEM1 ;$10->$18
EB11 10 80 8D       1617    DB $10,$80,$8D
EB14                1618  BUFF1
EB14 00 00          1619    DW 0
EB16 92 CF 87       1620    DB $92,$CF,$87
EB19                1621
EB19                1622  WTDMAD ;WRITE
EB19 11 83 79       1623   DB 17,$83,$79
EB1C                1624  BUFF2
EB1C 00 00          1625    DW 0
EB1E                1626  SLNG2
EB1E FF 00          1627    DW 256-1
EB20                1628  IOMEM2 ;$14->$1C
EB20 14 28 80 8D    1629    DB $14,$28,$80,$8D,$FB,$0F,$9
                          2,$CF,$05,$CF,$87
EB24 FB 0F 92 CF
EB28 05 CF 87
EB2B                1630
EB2B                1631  KENKO1 ;パッチ当て
EB2B ED 78 D9 ED    1632   DB $ED,$78,$D9,$ED,$79,$D9
EB2F 79 D9
EB31 ED 78 D9 02    1633   DB $ED,$78,$D9,$02,$00,$D9
EB35 00 D9
EB37 0A 00 D9 ED    1634   DB $0A,$00,$D9,$ED,$79,$D9
EB3B 79 D9
EB3D 0A 00 D9 02    1635   DB $0A,$00,$D9,$02,$00,$D9
EB41 00 D9
EB43                1636  KENKO5
EB43 CD 77 E9       1637   CALL DVINS
EB46 CD 83 E9       1638   CALL BFOUS
EB49                1639
EB49 D9 ED 78 D9    1640   DB $D9,$ED,$78,$D9,$ED,$79
EB4D ED 79
EB4F D9 0A 00 D9    1641   DB $D9,$0A,$00,$D9,$ED,$79
EB53 ED 79
EB55 D9 ED 78 D9    1642   DB $D9,$ED,$78,$D9,$02,$00
EB59 02 00
EB5B D9 0A 00 D9    1643   DB $D9,$0A,$00,$D9,$02,$00
EB5F 02 00
EB61 CD 6E E9       1644   CALL BFINS
EB64 CD 8D E9       1645   CALL DVOUS
EB67                1646
EB67                1647  REMMDT
EB67 D9 ED 78 D9    1648   DB $D9,$ED,$78,$D9,$ED,$79
EB6B ED 79
EB6D D9 ED 78 D9    1649   DB $D9,$ED,$78,$D9,$02,$00
EB71 02 00
EB73 ED 78 D9 ED    1650   DB $ED,$78,$D9,$ED,$79,$D9
EB77 79 D9
EB79 0A 00 D9 ED    1651   DB $0A,$00,$D9,$ED,$79,$D9
EB7D 79 D9
EB7F                1652
EB7F                1653  ;/ WORK AREA /
EB7F 00             1654  SID  DB 0
EB80 00             1655  TRN  DB 0
EB81 00             1656  SCT  DB 0
EB82 00             1657  LSCT DB 0
EB83 00 00          1658  RS   DW 0
EB85 00 00          1659  RE   DW 0
EB87 00             1660  LTR  DB 0
EB88 00 00          1661  RRC  DW 0
EB8A                1662
EB8A 00             1663  BSCS DB 0
EB8B 00             1664  BSCSWK DB 0
EB8C 00             1665  FDCCMD DB 0
EB8D 00 00          1666  DMACMD DW 0
EB8F 00             1667  EDD    DB 0
EB90 00 00          1668  CRSS   DW 0
EB92 00             1669  RCN    DB 0
EB93 00 00          1670  CSBYT  DW 0
EB95 00             1671  RIC  DB 0
EB96 00             1672  ED   DB 0
EB97 00 00          1673  SETOFF DW 0
EB99                1674
EB99                1675  ;
EC80                1676   ORG $EC80
EC80                1677  PATH
EC80 46 3A 2F 3B    1678   DM 'F:/;A:/;'
EC84 41 3A 2F 3B
EC88 00             1679   DB 0
EC89                1680
EC89                1681
EC89                1682  ;///////////////////
EC89                1683  ; DISK ACCESS
EC89                1684  ; for X1 (non DMA)
EC89                1685  ;///////////////////
EC89                1686
F000                1687   ORG $F000
F000                1688
F000 C3 06 F0       1689  #RWRECX1 JP X11
F003 C3 02 F1       1690  #WTTRCX1 JP X22
F006                1691
F006                1692  ;
F006                1693  X11
F006 3A 8B E0       1694   LD A,(#DN)
F009 FE 04          1695   CP 4
F00B D2 C5 E8       1696   JP NC,RAMRC ;RAMディスク
F00E                1697
F00E C5             1698   PUSH BC
F00F D5             1699   PUSH DE
F010 CD 78 E8       1700   CALL RECTRN
F013 38 73          1701   JR C,SSTPM@
F015 3A 81 EB       1702   LD A,(SCT)
F018 01 FA 0F       1703   LD BC,$FFA ;SCTレジスタ
F01B ED 79          1704   OUT (C),A
F01D 16 FB          1705   LD D,$FB ;DATA
F01F 1E F8          1706   LD E,$F8 ;STATUS
F021 D9             1707   EXX
F022 ED 4B 91 E0    1708   LD BC,(#BF) ;BUFF ADR
F026 D9             1709   EXX
F027 3A A1 E0       1710    LD A,(#FRWF)
F02A B7             1711    OR A
F02B C2 AD F0       1712    JP NZ,WTREC@ ;書き込み
F02E F3             1713   DI
F02F CD B3 E6       1714   CALL BGIOMM
F032 21 ED 79       1715    LD HL,$79ED ;OUT (C),A
F035 38 03          1716    JR C,PACX1  ;IF C GRAM
F037 21 02 00       1717    LD HL,$0002 ;LD (BC),A
F03A                1718  PACX1
F03A 22 A6 F0       1719    LD (REDPAC),HL ;パッチ当て
F03D 3E 80          1720   LD A,$80 ;コマンド READ
F03F CD E6 F0       1721   CALL SETSCT ;READY
F042 38 44          1722   JR C,SSTPM@ ;ERR
F044 CD 9B F0       1723   CALL RED1 ;READ
F047                1724  RWTOG ;共通ルーチン
F047 07             1725   RLCA
F048 E6 DF          1726   AND $DF
F04A 28 05          1727   JR Z,NDSTOP
F04C CD 31 EA       1728    CALL DSTOP ;ERR
F04F 18 37          1729    JR SSTPM@  ;
F051                1730  NDSTOP
F051 D9             1731   EXX
F052 60             1732   LD H,B
F053 69             1733   LD L,C
F054 ED 4B 91 E0    1734   LD BC,(#BF)
F058 B7             1735   OR A
F059 ED 42          1736   SBC HL,BC        ;
F05B ED 4B 9F E0    1737   LD BC,(#SLNG)
F05F B7             1738   OR A
F060 ED 42          1739   SBC HL,BC        ;大きさ
F062 D9             1740   EXX              ;CHECK
F063 28 08          1741   JR Z,NSSTP
F065 3E 01          1742    LD A,1 ;DATA量が変
F067 32 8C E0       1743    LD (#STOP),A
F06A 37             1744    SCF           ;
F06B 18 1B          1745    JR SSTPM@ ;ERR
F06D                1746  NSSTP
F06D 2A 9F E0       1747   LD HL,(#SLNG) ;
F070 ED 5B 91 E0    1748   LD DE,(#BF)    ;#BF=
F074 19             1749   ADD HL,DE      ;#BF+#SLNG
F075 22 91 E0       1750   LD (#BF),HL    ;
F078 2A 93 E0       1751   LD HL,(#REC) ;
F07B 23             1752   INC HL         ;#REC+
F07C 22 93 E0       1753   LD (#REC),HL ;
F07F 01 FC 0F       1754    LD BC,$FFC ;
F082 3A 8B E0       1755    LD A,(#DN) ;MTR OF
F085 ED 79          1756    OUT (C),A  ;
F087 B7             1757    OR A
F088                1758  SSTPM@
F088 01 D0 1F       1759   LD BC,$1FD0  ;
F08B 3A A2 E0       1760   LD A,(#WFD0) ;
F08E ED 79          1761   OUT (C),A    ;画面回り
F090 01 00 0B       1762   LD BC,$B00 ;
F093 3E 10          1763   LD A,$10   ;
F095 ED 79          1764   OUT (C),A  ;BANK
F097 FB             1765   EI         ;初期化
F098 D1             1766   POP DE
F099 C1             1767   POP BC
F09A C9             1768   RET
F09B                1769
F09B                1770  ;ディスクから
F09B                1771  ;読み込み
F09B                1772  RED1
F09B ED 78          1773   IN A,(C) ;STATUS
F09D 0F             1774   RRCA
F09E D0             1775   RET NC
F09F 0F             1776   RRCA
F0A0 30 F9          1777   JR NC,RED1
F0A2 4A             1778   LD C,D   ;DATAレジスタ
F0A3 ED 78          1779   IN A,(C) ;下位8BIT
F0A5 D9             1780   EXX
F0A6                1781  REDPAC
F0A6 ED 79          1782   OUT (C),A ;LD (C),A
F0A8 03             1783   INC BC
F0A9 D9             1784   EXX
F0AA 4B             1785   LD C,E ;STATUSレジスタ
F0AB 18 EE          1786   JR RED1
F0AD                1787
F0AD                1788  ;
F0AD                1789  WTREC@
F0AD F3             1790   DI
F0AE CD B3 E6       1791   CALL BGIOMM
F0B1 21 ED 78       1792   LD HL,$78ED ;IN A,(C)
F0B4 38 03          1793   JR C,PACX2
F0B6 21 0A 00       1794   LD HL,$000A ;LD A,(BC)
F0B9                1795  PACX2
F0B9 22 DF F0       1796    LD (WRTPAC),HL
F0BC 22 C0 F0       1797    LD (WRTPAC2),HL
F0BF D9             1798   EXX
F0C0                1799  WRTPAC2
F0C0 ED 78          1800   IN A,(C)
F0C2 08             1801   EX AF,AF'
F0C3 D9             1802   EXX
F0C4 3E A0          1803   LD A,$A0 ;コマンド WRITE
F0C6 CD E6 F0       1804   CALL SETSCT ;READY
F0C9 DA 88 F0       1805   JP C,SSTPM@
F0CC CD D2 F0       1806   CALL WRT1 ;書き込み
F0CF C3 47 F0       1807   JP RWTOG
F0D2                1808
F0D2                1809  WRT1
F0D2 ED 78          1810   IN A,(C) ;
F0D4 0F             1811   RRCA      ;STATUS
F0D5 D0             1812   RET NC    ;
F0D6 0F             1813   RRCA      ;
F0D7 30 F9          1814   JR NC,WRT1
F0D9 4A             1815   LD C,D ;DATA
F0DA 08             1816   EX AF,AF'
F0DB ED 79          1817   OUT (C),A
F0DD D9             1818   EXX
F0DE 03             1819   INC BC
F0DF                1820  WRTPAC
F0DF ED 78          1821   IN A,(C)
F0E1 08             1822   EX AF,AF'
F0E2 D9             1823   EXX
F0E3 4B             1824   LD C,E ;STATUS
F0E4 18 EC          1825   JR WRT1
F0E6                1826
F0E6                1827  ;READY
F0E6                1828  ;
F0E6                1829  SETSCT ;IN A
F0E6 32 8C EB       1830   LD (FDCCMD),A
F0E9 C5             1831   PUSH BC
F0EA D5             1832   PUSH DE
F0EB CD FC E6       1833   CALL MTRON
F0EE D4 6F E7       1834   CALL NC,SEEK
F0F1 D1             1835   POP DE
F0F2 C1             1836   POP BC
F0F3 D8             1837   RET C
F0F4 01 F8 0F       1838   LD BC,$FF8
F0F7 3A 8C EB       1839   LD A,(FDCCMD)
F0FA ED 79          1840   OUT (C),A ;コマンドレジスタ
F0FC 3E 07          1841   LD A,7 ;
F0FE                1842  SESCLP  ;
F0FE 3D             1843    DEC A ;WAIT
F0FF 20 FD          1844    JR NZ,SESCLP
F101 C9             1845    RET
F102                1846
F102                1847  ;/ WRITE TRACK /
F102                1848  ;       (non DMA)
F102                1849  X22
F102 C5             1850   PUSH BC
F103 D5             1851   PUSH DE
F104 CD 78 E8       1852   CALL RECTRN
F107 D4 FC E6       1853   CALL NC,MTRON
F10A D4 4D E7       1854   CALL NC,SEEKT
F10D 38 48          1855    JR C,SSTPM2@
F10F 3E 03          1856   LD A,3         ;仏の顔も
F111 32 5D F1       1857   LD (RTRY),A ;3度まで
F114 F3             1858   DI
F115 CD B3 E6       1859   CALL BGIOMM
F118 21 ED 78       1860   LD HL,$78ED ;IN A,(C)
F11B 38 03          1861   JR C,PACX3
F11D 21 0A 00       1862   LD HL,$000A ;LD A,(BC)
F120                1863  PACX3
F120 22 DF F0       1864    LD (WRTPAC),HL
F123 22 2F F1       1865    LD (WRTPAC3),HL
F126                1866  RTRYLP
F126 16 FB          1867   LD D,$FB ;STATUS
F128 1E F8          1868   LD E,$F8 ;DATA
F12A D9             1869   EXX
F12B ED 4B 91 E0    1870   LD BC,(#BF) ;BUFF ADR
F12F                1871  WRTPAC3
F12F ED 78          1872   IN A,(C)
F131 08             1873   EX AF,AF'
F132 D9             1874   EXX
F133 01 F8 0F       1875   LD BC,$FF8 ;FDC COMMAND
F136 3E F0          1876   LD A,$F0   ;WRITE TRACK
F138 ED 79          1877   OUT (C),A  ;
F13A 3E 07          1878   LD A,7 ;WAIT
F13C                1879  SESCLP2 ;
F13C 3D             1880    DEC A ;
F13D 20 FD          1881    JR NZ,SESCLP2
F13F CD D2 F0       1882   CALL WRT1 ;書き込み
F142 07             1883   RLCA
F143 E6 DF          1884   AND $DF
F145 28 0C          1885   JR Z,NDSTOP2
F147 21 5D F1       1886    LD HL,RTRY ;RETRY
F14A 35             1887    DEC (HL)   ;
F14B C2 26 F1       1888    JP NZ,RTRYLP
F14E CD 31 EA       1889    CALL DSTOP ;ERR
F151 18 04          1890    JR SSTPM2@
F153                1891  NDSTOP2
F153 CD 1F E7       1892   CALL MTROF
F156 B7             1893   OR A
F157                1894  SSTPM2@
F157 CD D9 E6       1895   CALL BGRST ;元へ戻す
F15A D1             1896   POP DE
F15B C1             1897   POP BC
F15C C9             1898   RET
F15D                1899
F15D 00             1900  RTRY DB 0
F15E                1901
F15E                1902
```

▶「20世紀梨ドリンク」もすごかった。「気分転館V」もすごかった。しかし，それをはるかに超える代物を知ってしまいました。それはUCCの「ワークス」シリーズです。なかでも「ハードワーク」は柿エキス入りというものでそれはもう……フタを開けた瞬間に死ねます。ほかの2種もはっきり言って毒物です。
升井　晋也（21）福岡県

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年7月18日の到着分までとします。当選者の発表は1990年9月号で行います。

シャープ　☎03(260)1161

1

ダウンタウン熱血物語

X68000用　5"2HD版2枚組

8,800円　3名

ファミコンでもお馴染みの熱血シリーズ移植第2弾。バリバリの硬派，くにお君とりき君が町じゅうを駆けめぐり大活躍する，ケンカアクションゲームだ。

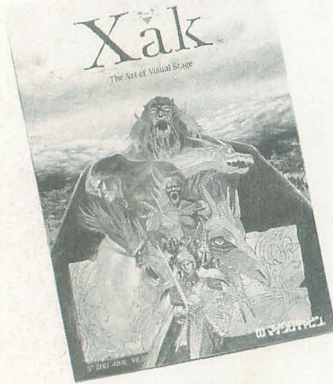
マイクロキャビン　☎0593(51)6482

2

サーク

X68000用　5"2HD版4枚組

8,800円　3名

ひさびさにX68000に新作を出したマイクロキャビンの自信作。主人公の青年が，悪の親玉バドゥーを封印するため旅に出るというRPG。

日本ファルコム　☎0425(27)0555

4

ファルコムシステム手帳

2名

ワンダラーズ・フロム・イースが評判の日本ファルコムから，システム手帳をプレゼント。2タイプ各1名ずつ，どちらが当たるかは着いてのお楽しみ！

システムソフト　☎092(752)3902

3

天下統一

X68000用　5"2HD版2枚組

9,800円　3名

大戦略などでお馴染みのシステムソフトの最新戦国シミュレーションゲーム。移植は同じ九州のよしみか，あのアルシスソフトが担当している。

日本ソフトバンクのメモ帳

5名

5

Oh!X編集部ご用達のメモ帳。このメモ帳，システム手帳にもはさめるというスグレモノ。5名の方にプレゼントしちゃいます。

5月号プレゼント当選者

1ポピュラス（秋田県）田口勝己（滋賀県）森夕香（奈良県）澤克彦　2バブルボブル（東京都）白水健太（茨城県）唯野晃（兵庫県）木村健二　3リウィード（北海道）渡部一郎（千葉県）中村敏祐（愛知県）白井達広　4ファルコムグッズ　A（静岡県）渡辺直樹（鹿児島県）本真光　B（秋田県）尾形淳一（東京都）阿部亨（長野県）塚本隆司　C（奈良県）川上峰且（兵庫県）石崎義忠　D（青森県）沢田優（宮城県）小野寺光（大阪府）堤博之　E（千葉県）大古哲生（香川県）森洋幸　5高麗人参飴（茨城県）安部博幸

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

（価格はすべて消費税別です）

［第2話］

本命は恐ろしい

TAKAHARA HIDEKI 高原 秀己

競馬を楽しむ女性が増えている。

純粋にギャンブルとして興じるいわゆる「オジンギャル」をはじめ，タケ・ユタカ人気に乗り遅れまいとする女の子，果てはデートコースとして馬券を買う彼にくっついてくる彼女，とパターンはさまざまだが，本当に競馬場でも場外馬券売り場でも女性の姿が目立って多くなった。

券を買うわけでもないのに彼にくっついてきて長蛇の列をさらに長くする女の子にはご遠慮いただきたいが，殺伐としたギャンブルの場に女性の姿が目立つ光景はいいものだ。

もともと華やかな競馬。見る側にファッショナブルな要素が加わって，オシャレなギャンブルとして人気急上昇。一種のブーム現象にまでなっているといっていい。

沸騰したのが5月27日に開催された“競馬の祭典”日本ダービーだ。東京競馬場の入場者19万人、JRA の売り上げ397億円といずれも競馬史上最高の活況にわいた。

4歳サラブレッド馬だけの戦いとあって，ほかのビッグレースとはひと味違う趣がプンプン。レースも最後まで大混戦となり，3番人気のアイネスフウジンが新記録タイムで今年のダービー馬に輝いた。武豊騎乗の人気馬ハクタイセイはゴール間近で足が伸びず，2着には本命のメジロライアンが入った。

連勝複式は3－5で，7.7倍。けっこう好配当だが，これ，2番目に人気があった組み合わせ。あとはすべて10倍以上のオッズだったから，大本命みたいなもの。穴を狙ったぼくも押さえに少しだけ買っておいたので，差し引きで若干のプラスになった。

それにしても，ビッグレース続きだったこの春の競馬。天皇賞からダービーまで毎週毎週，よくまあこれだけ飽きもせず本命格の馬ばかりが入ったもんだ。天皇賞のスーパークリーク，安田記念のオグリキャップはいうに及ばず，桜花賞，さつき賞，ダービーと終わってみればことごとく人気上位3頭のどれかが軒並み1着，2着を占めている。

よく飲みにいく店の常連サンに，

「競馬は夢を買うもんや」

という哲学を持った危ない風貌のおニイサンがよく来る。何百倍という大穴ばかりを狙い続けているが，戦績はサッパリ。こうなると，救いがない。キッチリと本命に賭ける地味な生き方のオジサンは逆に毎週連勝に次ぐ連勝。

ところで競馬の本命優位は一時的な現象なのかもしれないが，世の中全般にわたり，最近，まず「大穴」はこない。「予想外のハプニング！」なんてことはついぞお目にかからない。

前回は「選択の時代」という歓迎できる風潮になってきたことを書いたが，これは一面では自民党対社会党，トヨタ対日産といった感じの複数の大手が拮抗してそのフィールドを支配してしまうという大いに嘆くべき時代の到来を意味しているのかもしれないのだ。

プロ野球で考えてみると，はっきりわかるが，西武と巨人が独走していることに対し，奇妙な安心感が漂っている（ぼく自身は巨人がダントツの実力を持っているとは思っていないが，カープと並んで戦力が整っていることだけは認める)。阪神やダイエーが無人の黒星街道の野を疾走していることをすごく自然なものとして感じている人が多いのではなかろうか。

パソコン商戦もそう。このほど各社の'89年度販売実績と'90年度計画がまとまったが，トップメーカーである NEC の販売台数は'89年度ついに100万台に到達。'90年度は120万台が目標と，もうとどまるところを知らない。販売金額も'90年度にはとうとう5000億円を突破するというからものすごい勢いだ。

2位グループ（エプソン，東芝，富士通，IBM）以下のメーカーとの差はますます開いてきたし，この2位グループとそれ以下との格差もこれまた大きくなってきた。起死回生の下位メーカー連合といわれた AX パソコングループはいつしか存在自体が風前の灯に。AXとは，APPENDIX（つけたし）の略語だったのかしら？　という嫌みすら思いついてしまう有り様だ。

だけどこれ，すごく自然に感じてしまってそれがどこか心地いいのが不思議なところなのである。

コンピュータビジネス自体の大型化，寡占化という宿命にもよるところは大きいが，もはやソードのようなベンチャーの出る余地は全くないといってよかろう。

そういえば，「ベンチャービジネス」という言葉が新聞紙上をにぎわせること自体がめっきり減った。内需拡大にともなう鉄鋼，建設，産業機械など従来型ビッグビジネスの活況が連日，報じられている。

数年前とは明らかに違う現象だ。こんなことはなかった。社会構造の変化にともない，微妙に風潮まで変わってきていることは指摘できる。すべて同一線上に積み重なっているストーリーとして考えると乱暴かもしれないが，おもしろい作業ではある。

だけど。

そうなると，安定社会というレールの上を走るだけの時代に逆戻りしてしまうのではないか，という不安を感じてしまう。

塾，一流高校，一流大学から官庁や大手企業に入り，いい家柄のお嬢サマと結婚して子供をレールの上に乗せる。

一瞬ゾッとするが，ふと現実を見ると，すでにそうなってしまっているのだ。

まるで'60年代高度成長期のデ・ジャ・ブーを見ているような気がするのだが，実際に「岩戸景気」をしのぐ好況を迎え，似た雰囲気が各方面でいろいろと出はじめているという。

これで「階級」の概念が出てくると終わりなのだが，これまた実は出てきている。

土地暴騰にともなう資産格差という経済的階級制フレームワークは実は暗黙のうちに土地暴騰にともなう資産格差によって，出来上がってしまっている。しかも東南アジアからの労働者の台頭によって，より鮮明になってきているのだ。

恐ろしき本命主導時代。

歴史が繰り返すとするならば，オイルショックに似た経済破綻が襲ってくるはずなのだが。

さて。

ついに登場！X68000SUPER-HD

Tan Akihiko 丹　明彦

1990年X68000ラインアップの最上位機種，チタンブラックのX68000SUPER-HDの出荷が始まった。概要は5月号でお伝えしているが，このほど編集部に届いた製品をもとに試用レポートをお届けする。

どこが新しいのか

PROIIやEXPERTIIも1990年バージョンである。基本的には従来機種の設計を踏襲しつつ，SX-WINDOWを装備してなおかつ値下げを行っている(偉い！)。これに対し，SUPER-HDは，もちろん基本仕様こそ変更はないが，PROやEXPERTとは一線を画した設計になっているといってもいい。

SUPER-HDの目玉は，なんといっても，

SCSIを標準装備した

ことであろう。SCSIとはハードディスクや光磁気ディスクなど大容量外部記憶装置ほかさまざまな周辺機器に関する世界共通規格のひとつであり，今後外部記憶装置の仕様はSCSI規格に統一されていくものと思われる。

X68000の従来機種のハードディスクはSCSIに近いものではあるが純正のSCSI規格ではない。そのため，X68000にSCSI規格対応のハードディスクや光磁気ディスクをつなごうとすれば，SCSIインタフェイスを新たに接続する必要があった。

これに対し，SUPER-HDはSCSI規格の80Mバイトハードディスクを内蔵している。シークタイムも短く，クラス最高速の部類に入る。というわけで，OSもこれまでのものはそのままでは使えない。といってもHuman68k本体は基本的に同じで，新しくSCSI用のデバイスドライバを登録するようになっている。

SCSIデバイスサポート

したがって，ハードディスク周辺に関してだけ従来機種との互換性がない。まあ，ハードディスクは持ち歩いたりする性質のものではないので不都合はないだろう。また，いうまでもないことだが，フロッピーディスクへのアクセス関係には一切変更がないので，市販ソフトが使えないのではないかといった心配も無用である。

大容量ハードディスクや光磁気ディスクなど，今後メジャーになっていくであろう外部記憶装置を自由に扱えるようにするためにはSCSI規格への移行が必要である。今回の仕様変更は将来への布石なのだと見ることができよう。

ハードとともにソフトも少し変わる

X68000シリーズの常として，ソフトウェアレベルでの互換性は新製品でも完全に保たれている。ただSCSI対応になったため，ハードディスクなどに直接関わりそうな部分は少々変更されている。システムディスクのラベルを見ても，「SCSI対応」の文字が輝いている。変更になったソフトウェアのうちで主なものは，

COMMAND.X
FORMAT.XやDISKCOPY.Xなど
X-BASIC

といったところである。システム自体を始め，軒並みバージョン2.02，2.10というものに変わっている。

DRIVEコマンドを使うとBASICでのディレクトリ表示時にフロッピーディスクがチェックされるというバグも今回フィックスされているようだ。

ほかにも，ちょっと大きなところとして，ビジュアルシェル(VS.X)が付属していないことが挙げられる。EXPERTIIやPROIIにはちゃんとあったことを考えると，ビジュアルシェルがSCSI規格に対応できていないということであろう。

おそらく開発側ではCOMMAND.Xなどのようなバージョンアップは不要だと判断したのだろう。というのも，SX-WINDOWがビジュアルシェルの機能を十分カバーできているからである。だから困ることはまずない。今後はSX-WINDOWが主流になっていくということでもあるのだろう。

さて，SCSIドライバのほかにも，デバイスドライバの中には多少バージョンアップされたものがある。タイムスタンプを見る限りでは，

OPMDRV.X
FLOAT2.X

の2つが新しくなった模様である。こちらは，SCSI規格対応とは関係ないバージョンアップであろう。

もちろん，これまでの機種同様，WP.X(標準ワープロ)も付属するが，残念ながらワープロまわりはバージョンアップされていない。

もちろんマニュアルだって変わる

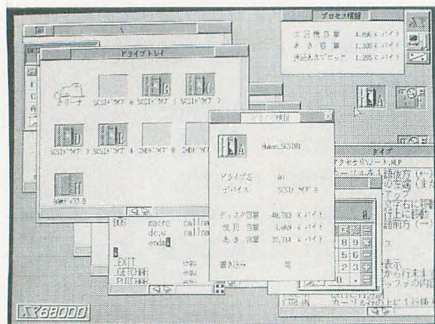
ビジュアルシェルがなくなり，SX-WINDOWが入った分，マニュアルも変わった。内訳は，

・取扱説明書
・Human68k ver2.0ユーザーズマニュアル
・SX-WINDOWユーザーズマニュアル
・X-BASICユーザーズリファレンス
・日本語ワードプロセッサ・辞書ユーティリティユーザーズマニュアル

となっている。

VS.Xがなくなったため取扱説明書の内容が全面的に改められたほかは，SUPER-HDだからといって，内容は特殊なものはない。PROIIやEXPERTIIにもほぼ同様のマニュアルが付属している。

X68000の良さは外観にもある

1990年バージョンは，EXPERTII，PROII，そしてSUPER-HDともに，本体の「X68000」のロゴがメタルエンブレムになって

いる。この外観にこだわるあたりがX68000魂を思わせてなかなかよい。

本体デザインは初期型からのマンハッタンシェイプ。キーボードもマウス・トラックボールも初期型からの伝統。ポップアップハンドルも基本である。本体色がチタンブラックというのは好き嫌いの分かるところだろう。

ディスプレイは例の音声多重(耳つきともいう)。チタンブラックのディスプレイはとりあえずこの1機種だけなのだが，高解像度の最上位機種なので，ディスプレイに関しては選択の余地は不要だろう。そうそう，21型ディスプレイもステレオになったのでよろしく。

広い広い80Mバイト

ハードディスクは買ったときのまっさらな状態では使えない。領域確保をしたあと，システムを入れて初めて機能するものである(物理フォーマットは初めからかけてある)。ハードディスクを使うことは，多少の予備知識が要求されることなのだ。ことにSUPER-HDの場合はハードディスクの容量が大きいので，システムの構築のしかたにユーザーのセンスが表れるといってもいいだろう。

ちょいと細かい話になるので恐縮だが，EXPERTやPROに付属のHuman68k (バージョン2.0)では，40Mバイトを超える容量の外部記憶は40Mバイトずつに分けて扱う必要があった。たとえば80Mバイトのハードディスクなら，40Mバイトのハードディスクが2台あるつもりで使うことになっていた。SUPER-HDのSCSIドライバにはそのような制限はなく，80Mバイトのハードディスクは80Mバイト1台である。もし望むなら，「80MバイトのAドライブ」を作ることも可能になっている。

しかし，1台のハードディスクをひとつのドライブとしてしか使わない，そういう領域確保のしかたがあまり利口ではないことも事実である。では利口な領域確保とはなんだろう。1台のハードディスクに複数のドライブを同居させるのが賢いとされている。実例を挙げると，たとえば編集室のSUPER-HDの場合，システムドライブに40Mバイト，辞書のドライブに5Mバイト，作業用のドライブを2つ作ってそれぞれ10Mバイト取り，プログラムには15Mバイトの領域を確保している。

このように，用途別にドライブを分けておくのは大容量ディスクを使う際の常套手段である。ハードディスクは，使い込むにつれて，膨大なファイルの情報が複雑に絡み合い，どうしても速度が少しずつ落ちてくるものだ。そんなドライブを整理するような場合でも，用途別のドライブを作っておくと実に便利である。とすると，この方法を採用する限り，80Mバイトだろうが40Mバイト×2だろうが使ううえではたいした差がないことになる(話に一貫性がないな)。

それから，ぜひとも驚いてほしいのは領域確保の速さである。なんと数秒で済んでしまう。従来機種が領域確保に分単位の時間を要していたことを考えると驚異的ですらある。不良セクタチェックを省略したのだろう。最近のハードディスクは丈夫なので領域確保なんてたまにしかやらない作業だが，それでも速いのはいいことだ。

かくのごとく広くて速いハードディスクであるが，それより速い記憶装置がある。それはメモリである。SUPER-HDのメインメモリはEXPERTと同じで標準で2Mバイトだ。標準システムがSX-WINDOWだとすると，まだこれでも足りないような気もする。

最後に

初代X68000が発表された当時，僕は両面高密度フロッピーディスク(HDと略するのだ)をハードディスクと勘違いした経験がある。そのときは「この本体にハードディスクなんか入るわけないじゃないか」と思ったものだった。ま，当時のハードディスクは20Mバイトでもけっこう大きかったし，X68000の本体からしてコンパクトなものだった(しかも縦置き)ことを考えると，ハードディスクの入る余地などないと考えたのも当然であろう。

ところが技術の進歩は実際たいしたもので，省スペース・高密度実装の結果，20Mバイトのハードディスクが載り，40Mバイトが載り，いまや80Mバイトである。

なんだかSUPER-HDの試用レポートといいながら，SCSIハードディスクの紹介に終始してしまったような気がするのだが，このことは，裏を返せば，X68000が頑固なまでにコンパチビリティを守り続けていることの証明でもある。初期型以来，どのモデルを使っていてもまったく違和感を感じない(もちろんメインメモリの容量やハードディスクの有無で使用感が左右されることはあるが)。

値段そのものはPROやEXPERTに比べると少し高いが，中身の充実度を考えると十分に安い。増設用のSCSIボードも発売されるが，特に将来光磁気ディスクを導入したいとお考えの方はSUPER-HDを第一候補にあげるのが賢明だと思う。

新しいSX-WINDOW

X68000SUPER-HDの発表はSX-WINDOWの正式な発表でもある。これまでもEXPERTII，PROIIにはSX-WINDOWが付属していたが，それ以前のX68000ユーザー用には近日発売予定のままで発売は開始されていなかった。今回，新しいバージョンがリリースされ，別売りバージョンの発売がいよいよ秒読み段階に入ったといえるだろう。

といっても，今回のSX-WINDOWでもファイル検索時のメッセージ表示が加わった程度しか目に見える違いはないが，内部では細かなバグフィックスや改良が行われていると思われる(当然?)。文字表示速度なども若干改善されているようだ。

システム以外の部分での大きな変更はウィンドウエディタ，ノート.Xに見られる。吉田幸一氏がコメントを差し控えた前バージョンと比べて，ウィンドウがカーソルと連動する，カット&ペースト機能が使えるなどの改良がされている。なぜか，コントロールキーのキー操作がバッファに溜まるという改悪点も見られるのが残念だ。そのほか，小さなところでは，暁子.Xの暁子さんの自転車に乗っているコンピュータがMacintosh(?)からX68000に変更されている。

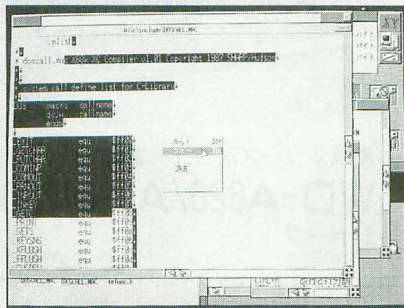
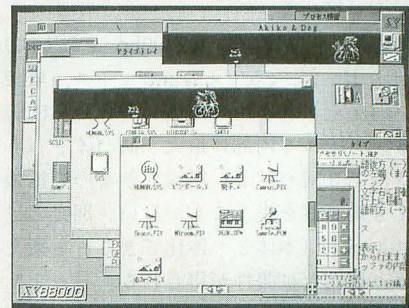
SX-WINDOW用のマニュアルは操作法と付属ファイルの解説に終始しており，ウィンドウ上でのアプリケーション開発用資料，および開発システムの公表はまだしばらく先になる模様。早くウィンドウ上のアプリケーションを開発したいという方にはもうしばらくおあずけだ。市販のアクセサリにも大いに期待したいところだ。

(S.N.)

カット&ペーストもできる

新しい暁子さん

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

情報ツール「All in Note」
AX286N-H2
シャープ

AX286N-H2

シャープはノート型のAXパソコン「AX286N-H2」を発売した。情報ツール「All in Note」と呼ばれるこのマシンには簡単に使いこなせるソフト「Business Mate」と，アプリケーションソフトの組み込みと起動が容易に行える「SPシステム」が標準装備され初心者にも使いやすくなっている。また，20Mバイトの容量を持つ2.5インチハードディスクを採用し，本体は厚さ34mm，重量2kgと小型かつ軽量である。

表示は冷陰極管バックライト（サイドライト方式）付きで鮮明な白黒表示のトリプルスーパーツイスト液晶ディスプレイにより鮮明で見やすい白黒表示（8階調）を実現。また，画面もAX仕様の640×480ドットと従来のノート型パソコンよりもひと回り大きくなっている。

さらに，オプションは2400bpsクラス5のモデムボード，CRTインタフェイス，増設RS-232Cなどの中から用途に合わせて内蔵でき，メモリも内部に最大2Mバイト（トータルで3Mバイト）増設できるので拡張しても携帯性を失うことがない。別売の拡張ユニット（本体と一体化できる）を使えばより豊富な拡張性が得られる。バッテリー駆動時間は約1.7時間で，一体型外部バッテリーを使えば5時間にできる。

価格は税別で398,000円。
〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(260)1161

UNIXオフィスプロセッサ
OA-120
シャープ

シャープはモトローラ製のMC68030を搭載したUNIXオフィスプロセッサ「OA-120」を発売した。このワークステーションは従来機種「OA-110WS」の2倍以上の性能を実現しており，OSには「UNIXシステムVリリース3.0」を採用している。また，高性能32ビットUNIXワークステーションながら，幅425mm×奥行き420mm×高さ120mmのコンパクトサイズを実現している。

「OA-120」ではハードディスクに加え150Mバイトのカセット磁気テープ装置が接続でき容易に大容量バックアップが行える。ほかにも液晶センサーパネルや，最大16台のPOSターミナルが接続可能である。

内蔵ハードディスクが40Mバイトと80Mバイトのものの2タイプがあり，価格はそれぞれ1,495,000円，1,795,000円(税別)となっている。
〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(260)1161

OA-120

書院シリーズ&ニュー書院
WD-A620/A720/7000
シャープ

シャープはパーソナルワープロ初のデュアルフォント&アウトラインフォントを内蔵した「書院」シリーズの新機種WD-A620/

WD-A720

WD-7000

720と，32ビットCPU搭載の「ニュー書院」WD-7000を発売した。

「書院」シリーズの新機種はパーソナルワープロとしては初の明朝体とゴシック体をROMで標準装備したデュアルフォントとアウトラインフォント機能，パーソナルDTP機能などの搭載により高度なレイアウト文書なども表現力豊かに美しく仕上げることができるようになった。プリンタは毎秒52字の高速印字で，高密度印刷のできる56/52/48ドットのものを採用している。また，辞書にはAI-V3辞書を搭載しており口語文変換などの変換効率を高めている。そのほかの機能としては電子手帳とデータの共有ができる電子手帳通信機能，個人データの管理に便利なパーソナルデータ管理機能などがある。さらに，WD-A720は上記に加えて3.5インチフロッピーディスクドライブを2基搭載（WD-A620は1基）しており，通信ソフトやMS-DOSコンバータなども標準で備えている。価格はそれぞれ145,000円と178,000円（税別）。

書院シリーズの最上位機種である「WD-7000」は32ビットCPUを搭載し処理速度を大

幅に向上している。「WD-7000」は40Mバイトハードディスク，そしてA4フルページ表示（24ドット）が可能な高解像度15インチ縦型CRTを内蔵，さらにA3サイズ対応の高解像度400DPIレーザープリンタを用意するなど高性能なビジネス用日本語ワードプロセッサとなっている。

また，矩形単位の移動・複写・切り取り・貼り付け機能による編集作業の効率アップ，アウトラインフォントおよび正楷書体標準装備による表現力の向上の実現なども果たしている。価格（税別）は標準システム（本体，CRT，キーボード，マウス含む）が1,208,000円，レーザープリンタシステムが2,150,000円。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(260)1161

CCITT規格V.42bis, 38400bps
Multi modem V32L
コア

コアは「Multi modemシリーズ」の新機種を発売した。「Multi modem V32L」はCCITT（国際電信電話諮問委員会）がモデム用データ圧縮方式のプロトコルとして採用した「V.42bis」を搭載している。CCITT規格V.42bis機能を搭載したモデムとしては日本で初めての発売となる。機能は一般回線用モデムとしてV.42エラー訂正，セキュリティコールバック，リモートコンフィギュレーションなどを装備し，V.42bisデータ圧縮機能により38400bpsの超高速を実現している。また，2線および4線専用回線にも利用でき4線専用回線に対する一般回線による自動バックアップおよび自動復帰機能を持っている。価格は298,000円。

〈問い合わせ先〉

㈱コア　☎045(441)8611

Multi modem V32L

Z80完全上位互換16ビットCPU
R800
アスキー

アスキーはザイログ社のZ80と完全上位互換性をもつ高速16ビットマイクロプロセッサ「R800」を開発した。「R800」は同一

R800

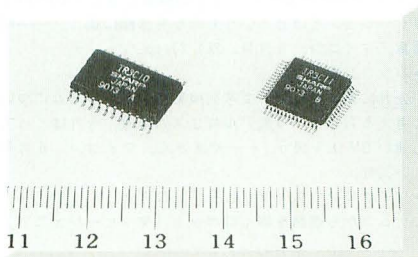
クロックのZ80と比較して3.6倍の高速動作を実現している。また，メモリ空間はメモリ拡張回路により16Mバイトまで拡張可能である。そのほかの特徴としては，高速な16ビット掛け算命令のサポート，2チャンネルのDMAコントローラ内蔵，7階層の優先順位をもつ強力な割り込み機能などがあげられる。

この「R800」はZ80互換マイクロプロセッサとして世界最高速の性能を実現している。このように高速化できた背景には，16ビットALUバスの採用，RISC CPUなどで用いられるパイプライン処理により1命令あたり平均1.3CPUクロックに実行速度が上がったこと，28MHzの高速クロック動作（CPUクロックは4分の1）の実現などがある。

そして，表面実装に最適な100ピンQFPに搭載することで従来の40ピンDIPと比べ体積比で3分の1以下の小ささを実現し，CMOSプロセスを使うことで低消費電力となっている。

〈問い合わせ先〉

㈱アスキー　☎03(797)6506

レーザーピックアップ駆動回路のIC化
IR3C10/IR3C11
シャープ

シャープは光ディスク用レーザードライバ「IR3C10」とレーザーコントローラ「IR3C11」を開発した。光ディスクは記憶容量の大きさなどから次世代の記憶メディアとして期待されているが，アクセスタイムの点ではまだ技術的には成熟しておらず，磁気ディスクに比べ劣っているといわれている。この要因のひとつとしてレーザーピックアップの大きさと，ピックアップスライドモーターの重量が考えられ，ピックアップ内に搭載される半導体レーザードライバ部のIC化が望まれていた。IR3C10およびIR3C11はこれに応えるものである。レーザーピックアップ部の小型・軽量化を考慮してコントロール部はピックアップ外に配置することを想定したため，あえて2チップ構成になっている。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(260)1161

OS-9/X68000用
SrcDbg V2.0
マイクロウェア・システムズ

マイクロウェア・システムズはOS-9/X68000上で動作するマイクロウェアCコンパイラ用のソースレベルデバッガSrcDbgを発売した。これを使用してCプログラムをインタラクティブに実行することにより，プログラムの開発時間を短縮することができる。また，アセンブリ言語レベルでもOS-9付属のDEBUGコマンドとほぼ同様にデバッグできる。パッケージ内容は，

ソースレベルデバッガ　SrcDbg
ヘルプファイル
練習用サンプルプログラム　Where.c
SrcDbgユーザーズマニュアル

となっていて，価格は39,800円。

〈問い合わせ先〉

マイクロウェア・システムズ㈱
☎03(257)9000

INFORMATION

シャープ ワールドサッカー'90
in 東京

オルドリッジ（レアル・ソシエダ）

レナト（フラメンゴ）

来たる8月6日（月）に東京ドームで「シャープ　ワールドサッカー'90」が開催される。この大会は昨年から行われているものであるが，本年度はブラジルの名門チーム“CRフラメンゴ”とスペインの強豪“レアル・ソシエダ”が来日して対戦する。チケットはすでに発売中。入場料金は自由席2,000円（大人），1,000円（小・中学生）から。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(260)1161

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。うっとうしい梅雨も終わりです。さあ，夏休みに向けて楽しい計画を立てましょう。補習のある人は別ですけど……。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
テクノポリス　徳間書店
POPCOM　小学館
マイコン　電波新聞社
マイコン BASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶ネットワーカー・ホリック第20回
アイドル情報が自慢のフジサンケイ系ネット「EYE-NET」と，集英社のコミック雑誌編集部直営のネット「BJネット」を紹介。PDSはPC-9801のゲーム「クアトロ」「タムタム」を紹介。——編集部，LOGIN，9・10号，214-215pp.

▶アルゴリズムを見切ったぞ!?
ブロック崩しのボールの動きを処理するアルゴリズムを例に，プログラムの組み方を解説している。——編集部，テクノポリス，6月号，100-104pp.

▶ディスクファイルのトラブル　注意点と復活対策
パソコンが広く使われている現在，そのデータの保守管理はますます重要になっている。ここではフロッピーをめぐる物理的・論理的なトラブルの内容とその対処法について考える。——佐田守弘，マイコン，6月号，138-151pp.

▶ MICOM COMEON GOODS
今月は，最近人気のコンピュータミュージック機器をまとめて紹介する。モニタスピーカーや MIDI 用ソフトウェアなどなど。——編集部，マイコン，6月号，158-163pp.

▶シャープ　春の見・体・験フェア
4月14・15日新宿 NS ビルで行われたシャープのフェアの模様をリポートする。——編集部，マイコン，6月号，166p.

▶シミュレーションウォーゲーム研究
X1turbo 用にも発売が決定している「大航海時代」について，光栄のシブサワ・コウ氏が解説。——シブサワ・コウ，マイコン，6月号，188-192pp.

▶ LET'S PROGRAM
今月の宿題は「階乗の計算」。オーバーフローを起こさぬようテクニックを凝らしたサンプルが X-BASIC，アセンブラ，X1用 BASIC で発表されている。——藤本健，マイコン，6月号，231-240pp.

▶ここまできた DTP の世界
日本でもサムシンググッド社から発売になった「アルダス・ページメーカー」。ソフトの概要と併せて DTP パブリケーションとはどういうものかを詳細に追う。——稲澤薫，マイコン，6月号，257-271pp.

▶パソコン横丁
先月に引き続いて，文字列検索のアルゴリズムについて考えるおしゃべり風アルゴリズム講座。今月はとっても速い BM 法を扱う。——やまさん，マイコン，6月号，300-304pp.

▶マイクロマウス　マッピージュニア
ナムコから迷路を解くロボット「マッピージュニア」が発売になった。その構造，パソコンをつないでの遊び方についての解説記事。——時国修，マイコン，6月号，321-327pp.

▶実践ハード入門
今月は磁気センサの製作。ホール素子を使って磁界の強さを測る。ミニ知識「電磁誘導」付き。——石川至知，マイコン，6月号，347-351pp.

▶ハードディスクであなたが変わる
最近普及の目ざましいハードディスクの入門記事。ハードディスクとは何かから始まって，その利点，種類，購入などについて述べる。——吉沢正敏，I/O，6月号，82-88pp.

▶東側ハイテク事情
東欧改革以後，西側のハイテク企業が続々と市場に参入している。その影響と未来についてのレポート。——ジョン・ヒルカーク，I/O，6月号，220-221pp.

MZ シリーズ

▶誌上公開質問状
MZ-80K/MZ-2000のカセットの修理を受け付けてくれるところは？　MZ-2500用のマシン語の本は？　などの質問に答えている。——PEGASUS，マイコンBASIC Magazine，6月号，90p.

MZ-1500(MZ-5Z001 BASIC)

▶Ｊ君の奇妙的冒険
柱をのぼっていく波紋修業ゲーム，あぶくをよける仙道ゲームの2つがミックス。——FROG，マイコン BASIC Magazine，6月号，125-126pp.

MZ-2500(M25-BASIC)

▶がんばれ！　孫五円
雲に乗ってぽんぽこ山に咲くガピタンの花をすべて摘む，パズルゲーム。——謎のパズル大好きおじさん，マイコン BASIC Magazine，6月号，127-128pp.

X1/turbo/Z

X1シリーズ

▶画像圧縮プログラム
X1用の画像圧縮プログラムを BASIC で。データ圧縮についての参考にもなる。——山本克稔，マイコン BASIC Magazine，6月号，73-74pp.

▶７殺し
７並べの変形版カードゲーム。——ポケタ玉井，マイコン BASIC Magazine，6月号，155-157pp.

▶ FRONTIER
20年間でどれだけ街を発展させられるかを競う。都市開発ゲーム。——パル，マイコン BASIC Magazine，6月号，158-160pp.

新刊書案内

デザインに戸田ツトムをフィーチャーしたNTT出版の BOOKS IN ▶ FORUM シリーズは情報・コンピュータ関係の怪しげで面白そうな本を出してくれる。今回の「情報の歴史」もまた格別である。枕許に置いて，いつでも手に取って眺めていたい。

この本はオールカラー写真ナシのただの年表である。タイトルにある「情報」は“人間が獲得し改良しつづけてきたコミュニケーションのためのすべての手段および内容。”(「関係の発見のために」より）のことである。そういったことであるから，紀元前7000万年前から1990年までの壮大な年表だ。むろん，1990年は白紙。教科書の副教材にあるような退屈な年表と思ってはいけない。独自の切り口で，文字情報だけでありながら平板でない面白い年表。これを作成する労力と知力を考えるとぞっとするくらいだ。この本は歴史を調べるためにあるのではなく，眺めながらいろんな発見をするためにある。「情報」をおいかけることは人の「文化」を見ることであり，「コミュニケーション」の歴史をつかむことである。高いけど，値段分の価値はあるぞ。（K）

情報の歴史　松岡正剛監修　編集工学研究所構成
NTT 出版
☎03(435)1212　A4判　433ページ　4,800円

書籍の価格は消費税込みです

X1+FM音源ボード(要NEW FM音源ドライバ)
▶ THE SUPER 忍 LONG DISTANCE
メガドライブ版ザ・スーパー忍のミュージックプログラム。——木村直之，マイコンBASIC Magazine，6月号，187-188pp.

X1turboシリーズ
▶ NEW SOFT
セレクテッドソーサリアンを紹介。——編集部，LOGIN，9・10号，24p.
▶ How To Win
セレクテッドソーサリアンを紹介。——編集部，コンプティーク，6月号，132-135pp.
▶先取りおすすめゲーム
6月発売予定の大航海時代を紹介。——編集部，テクノポリス，6月号，8-11pp.
▶月刊ソーサリアンニュース
セレクテッドソーサリアンの妖夢幻伝説，闇に消えた女神を紹介。——編集部，テクノポリス，6月号，26-27pp.
▶ AIR FORCE
弾を撃ちあい，相手のヘリを落とす。2人用。——山田仁，マイコンBASIC Magazine，6月号，161-162pp.

X68000

▶激突!! 対戦ゲームが新しい
近頃流行りの2人対戦可能のゲームで遊ぶ。X68000用は対戦ポピュラスを紹介。——編集部，LOGIN，9・10号，134-137pp.
▶ X68000新聞
新着ゲームの「ウルティマV」，「ブレード オブ ザ グレートエレメンツ」，アイデアプロセッサハイパーテキストワープロ「Hyperword」，バージョンアップしたサイクロン「サイクロンα68」の紹介。SX-WINDOW続報。——編集部，LOGIN，9・10号，148-153pp.
▶最新ゲーム徹底解剖!!
ポピュラス紹介の第3回と，グラナダ。対戦ポピュラスと，自分で面をエディットするカスタムモードを解説。グラナダはステージ1から3を攻略——編集部，LOGIN，9・10号，184-187pp.
▶ X68000SPIRITS
ポピュラス，FAR SIDE MOON，サークの紹介と，プロテニスワールドコート，ルーンワースの移植情報。——編集部，コンプティーク，6月号，240-241pp.
▶先取りおすすめゲーム
7月下旬発売予定のウルティマVを紹介。——編集部，テクノポリス，6月号，12-15pp.
▶ GAMING WORLD
新作&移植情報。戦略シミュレーションゲーム「JOSHUA」，ファンタジーアクション「バルーサの復讐」を紹介。——編集部，テクノポリス，6月号，30-32pp.
▶ WE ARE THE X68000 WORLD
新着ゲームと発売予定のゲーム「ラグーン」「ダウンタウン熱血物語」「スライミャー」「バルーサの復讐」「クォース」「FAR SIDE MOON」を紹介。——編集部，POPCOM，6月号，80-83pp.
▶誌上公開質問状
SYSファイルの転送のやり方，PATH指定の方法，マシン語を始めるのに必要なソフトは？ などの質問に答えている。——多田太郎，マイコンBASIC Magazine，6月号，89-90pp.
▶ HOT INFORMATION
X68000ハイパーテキストワープロ「Hyperword」を紹介。——編集部，マイコンBASIC Magazine，6月号，91p.
▶ SORCERESS
一人前の魔女になるために修業の旅に出たあなた。ほうきに乗って魔物を倒そう。——林純一，マイコンBASIC Magazine，6月号，163-164pp.
▶ DROP BLOCK
テトリス+オセロ!? 対戦パズルゲーム。——永井崇博，マイコンBASIC Magazine，6月号，165-167pp.
▶ゲームレビュー
アルシスソフトの「ナイトアームズ」を取り上げる。斬新な試みに高い評価を与えているが，熟成度はイマイチか？——あゆかわさつみ，マイコン，6月号，198-199pp.
▶ X68000マシン語入門
グラフィック関係のシリーズ第2回。グラフィックを加工する便利なプログラムを4本収録。色分布の表示や平均化など。——高橋雄一，マイコン，6月号，338-346pp.
▶なんでもQ&A
「辞書ディスクをRAMディスクに"DISKCOPY"しようとしたらできなかった。なぜ？」「X68000で使えるカラーインクジェットプリンタは？」などの質問にシャープの人がバッチリ答えちゃうぞ。——編集部，マイコン，6月号，408-409pp.
▶ SOFTBOX
シャープから発売になった「Hyperword」。その概要を紹介する。——L&M，I/O，6月号，126-127pp.
▶ Disk Optimizer
使いこんで不連続クラスタばかりになってしまったディスクを整理するユーティリティ。作業用ディスクが不要，HDにも対応していることなどがウリ。——高城英誌，I/O，6月号，144-154pp.
▶常駐HELP
いつでもドキュメントファイルが呼び出せるというユーティリティだ。ASCIIコードの表示・拡張ファンクションキーなども使えるぞ。——ずん，I/O，6月号，155-159pp.
▶ GAME BOX
メタルサイトとワンダラーズ・フロム・イースのレビュー。——市原昌文・吉沢正敏，I/O，6月号，166-167pp.
▶ YE
以前掲載されたフルスクリーンエディタの最終バージョン。細かい修正とバグ・フィックスが行われている。——井本祐司，I/O，6月号，177-187pp.
▶画面セーブプログラム
割り込み型の画面セーブプログラム。メニューでのオプション選択もできるので，フラクタクル作成中の画面退避も可能。——原道宏，I/O，6月号，200-205pp.
▶白黒ビデオ画像入力装置の試作・実験
イメージユニットは高い！ という人のために，イメージユニット用インタフェイスを使って自作の画像入力装置をつなごうというわけ。——ペリカン君，I/O，6月号，250-255pp.
▶ AVプログラミング講座
X68000を例にして，グラフィックイメージの拡大・縮小・回転と，ラスタスクロールのアルゴリズム，そしてそれらの応用について解説する。——中山進・仲田津宏，ASCII，6月号，249-256pp.
▶ WSEL
OS-9/X68000用のファイルセレクタ。ただしパーソナルウィンドウシステムが必要。——野出久司，ASCII，6月号，328-329pp.
▶ AV STRASSE
EXPERT II/PRO IIから標準添付されたSX-WINDOW Ver.1.0と，Console/GraphicsIOCS Ver.1.0の試用レポート。——仲田津宏，ASCII，6月号，345-347pp.

ポケコン

PC-E500
▶ GHOST CATCH
2～4人用ゲーム。ひとりがゴーストになり，残りのメンバーがそれを追う。——せとけん，マイコンBASIC Magazine，6月号，173p.
▶だいちゃんの日記 ～バレー編～
対コンピュータ・バレーボールゲーム。相手は4人から選べる。——神谷栄治，マイコンBASIC Magazine，6月号，174p.
▶ポケットコンピュータ活用研究
先月扱った表計算プログラムを使って，PC-9801の「Lotus 1-2-3」とデータをやりとりする。データファイルの形式や，データ交換の手順など。——塚田洋一，マイコン，6月号，328-334pp.

作品としてのプログラム

プログラミングやプログラマの実態，心得，考え方，知識。著者の基本は，プログラムは人間が作る人間臭いもの，ということだ。読み物としてもいいが，ちゃんと読めばプログラミングや計算機の勉強にもなる。日本語とプログラム，モデル化からオブジェクト指向，人工知能まで幅広く鋭く鋭く語られている。プログラミングを学ぶことは言語修得やテクニック修得では決してない。長生きできるプログラマの条件付。(K)

黒川利明著　岩波書店　☎03(265)4111　B6判　199ページ　1600円

バックランドでつかまえて

東京トンガリキッズをご存じか？ あれはトンガリキッズの物語の断片集だった。これもそう。ゲーセンをめぐるゲームキッズの物語の連作掌編集。それがどうだ。面白いではないか。ゲーセンをひとつの文化として育った私たちには，イタリア製のおシャレなアイテムを並べるよりクレイジークライマーをめぐる物語が妙にしっくりくる。そして，これだけゲームと人間の関係，ゲーマーの気持ちを正しく描いた物語はほかにはない。(K)

田尻智著　JICC出版局　☎03(234)4621　B6判変形　199ページ　900円

書籍の価格は消費税込みです

QUESTION and ANSWER

現在マシン語のIPL起動ディスクを作ろうとしているX1turbo IIユーザーです。いろんな書籍を参考にしてできた！　と思ったのですが，やはりダメでした。

同封したディスクを見てもらえばわかりますが，僕のIPL用ディスクの考えは，

・レコード０のFCBデータ，14のFATデータ，16のディレクトリ，32以降のメインプログラムがあればもう十分！　そしてこのディスクを立ち上げると，画面の左上に，「亜」の１文字が浮かび上がるんだ！　と思っていましたが結果はやってみてのとおりです。

いったいどこが間違っているのでしょうか。適切なアドバイスよろしくお願いいたします。　　青森県　今　隆範

質問の意味がよくわからないと思いますので，ちょっと補足説明させてもらいます。今さんが送ってくれたディスクには，TEST.Sysというファイルが入っていました。このファイルが「亜」の１文字を画面の左上に表示するプログラムになっていて，IPLでこのプログラムを読み込んで実行すれば「亜」が画面に浮かび上がる……となるわけです。ところが，実際にはまったくうまくいかない。

だからといって質問の中にも出てくる今さんのIPLに対する考え方が間違っているのかというとそうではないのです。これにはいくつかの理由が考えられます。なによりまずプログラムにバグがあることを疑ってみる必要がありますが，この点については，正常に動作することがあっさり確認できました。で，このときわかったのですが，このプログラムはIPLで読み込みに相当時間がかかっていたにもかかわらず，実際には100バイトにも満たない小さなものだったのです。となるとIPLにするファイルの大きさなどの情報が書かれている場所，すなわちレコード０に書かれている情報に間違いがある，もしくはFATが破壊されている可能性が大です。

まずはレコード０の内容を疑ってみることにします。リスト1-Aが今さんが送ってくれたディスクのレコード０の内容の一部をダンプリストにしたものですが，これだけ見てもどこが悪いのかまったくわからないことでしょう。そこで正しいレコード０の作り方を話してみようと思います。

IPLから起動することのできるプログラムはマシン語ファイルだけと決められていますから，適当なマシン語ファイルをIPLで起動するように変更してみましょう。

リスト1-Bを見てください。これは今回の説明のために$C000_H$～$C1FF_H$の範囲に実行可能なマシン語プログラムがあることを想定してディスクにセーブし，そのディレクトリのダンプリストをとったものです。

ひとつのファイルがディレクトリ領域に32バイトの情報を持っているということは皆さんご存じでしょう。この情報を手掛かりに，レコード０に書き込むデータを作成することになります。ひと目見て２～17バイトにあるのがファイル名だとわかりますが，先頭の１バイトはなんなのでしょうか。

これはファイルの種類を表すもので，第０ビットが１になっているものがマシン語ファイルであることを表します。IPLにするプログラムはモードが01_Hでなくてはならないので，第０ビットが1であっても41_Hなどの場合は01_Hに変更する必要があります。さらにファイルの拡張子は「Sys」でなければならないので，そのように変更します。

先に進んで31バイト目はファイルの格納されている先頭クラスタの位置を示しますが，IPL起動にする場合はこの部分をレコードの位置で表すことになっています。ですから，１クラスタは16レコードだから，

$5 \times 10_H = 50_H$

を書き込めばいいのです。

たいていのマシン語ファイルはレコード０の先頭から32バイト，この２つについて変更したものを書き込んでおけば，IPL起動に変更することが可能なんです。

このことを頭に入れて，今さんがIPL起動にしたいファイルのディレクトリ（リスト1-C）を見てみると19，20バイトの値がちょっとおかしいようです。この部分はファイルの大きさを表す部分で，下位，上位バイトの順で表すことになっていますが，今さんはこの値をディスクエディタなどで直接書き込んだのでしょうか？　上位下位を逆にしたために，本来38_Hバイトのはずのファイルを3800_Hバイトとなっています。レコード０にも同じように書かれているので，IPLはこのファイルの大きさを3800_Hバイトだと思い込み，それゆえ起動に時間がかかっていたのです。

ところが，この値を逆にすれば万事解決と思いきや，まだうまくいかない。ファイルを格納する先頭アドレスと実行アドレスまで考える必要がありそうです。ディレクトリでは21，22バイトに格納先頭アドレス，

リスト１

A

```
:C000=01 54 45 53 54 20 20 20 20 20 20 20 20 20 53 79 /.TEST          Sy
:C010=73 20 00 38 00 00 00 00 00 00 00 00 00 00 20 00 /s .8.......... .
```

B

```
:C000=01 44 45 4D 4F 20 20 20 20 20 20 20 20 20 4F 62 /.DEMO          Ob
:C010=6A 20 00 02 00 C0 00 C0 89 52 15 23 53 00 05 00 /j ...ﾀ.ﾀﾙR.#S...
```

C

```
:C000=41 54 45 53 54 20 20 20 20 20 20 20 20 20 53 79 /ATEST          Sy
:C010=73 20 00 38 00 00 00 00 00 00 00 00 00 00 02 00 /s .8............
```

D

```
:C000=01 54 45 53 54 20 20 20 20 20 20 20 20 20 53 79 /.TEST          Sy
:C010=73 20 38 00 00 80 00 80 00 00 00 00 00 00 20 00 /s 8.._._...... .
```

23，24バイトに実行アドレスを，いずれも下位，上位バイトの順で表すことになっていますので，このファイルは格納先頭アドレス，実行アドレスともに0番地となっていることがわかります。はたして0番地から実行されるものであってよいのか疑問だったのですが，逆アセンブルしてみるとリロケータブルファイルであることがわかったのです。「リロケータブルファイル」とは，どんなアドレスにプログラムが置かれていても実行することのできるファイルのことをいいます。

となると，どこの具合が悪いのでしょうか。それはこのプログラムがBIOS ROMを使用していることに起因しています。周知のとおりBIOS ROMを使うプログラムは8000H以降に置かなくてはならないことになっていますから，格納先頭アドレス，実行アドレスを8000Hに変更して起動すれば……やっとうまくいきました。これでFATが破壊されているという疑いは消えたことになります。結局リスト1-Dが正しいレコード0のかたちです。

X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが，WP.Xでは48ドット印字ができませんし，ほかの方法では昨年の7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。

それからWP.Xの環境設定の部分などをいじって文字をビットイメージ出力でなくコード出力にして，48ドット文字を印字するようにできないでしょうか。

兵庫県　中田　勝啓

WP.Xを起動して画面右上の「その他」をクリックしてみてください。プルダウンメニューの書式設定を選択すると，書式設定のウィンドウが開いたでしょう。その中の文字間に注目してください。文字間には密着，狭い，普通，広い，と4種類ありますが，これを密着にすれば複数文字にわたるロゴなどを定義した外字をくっつけて印字することができます。

図1

Q　X68000とCZ-89C4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法ではOh!X昨年の7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。
それからWP.Xの環境設定の部分などをいじって文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。
兵庫県　中田　勝啓

図2

Q　X68000とCZ-89C4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法ではOh!X昨年の7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。
それからWP.Xの環境設定の部分などをいじって文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。
兵庫県　中田　勝啓

ところが，この設定は困ったことに文書全体にわたって効果が出てしまうので，長い文書を印字するときは文字がくっついて読みづらいことになるでしょう。そのために密着部分だけあとで印字するとか，ユーザー自身が工夫しなくてはとても使いものにならないでしょう。PC-9801用のワープロでは文字単位に密着指定できて当たり前となっています。

図1が通常の出力，図2が密着指定をしたときの出力です。これでは文字が詰まりすぎなので，文字間に半角スペースを挟んで調整したものが図3です。これは，ファイル出力した文書をエディタで読み込み，

ESC・@
カーソル右
スペースキー
UNDO
（しばらく待つ）
ESC

のように操作します。

さらに，手作業で調整すれば図4のようになります。

それからWP.Xの出力をコード印字にすることができるのか，という質問ですがこれは無理です。どうしてもコード印字をしたいのなら，4月号の質問箱でとりあげたようにWP.Xの印刷機能を使わないで，自作のプリントアウトプログラムを使うようにすればいいでしょう。確か，以前電脳倶楽部にも48ピンプリンタ対応の印刷プログラムが載っていたと思います。（影山　裕昭）

図3

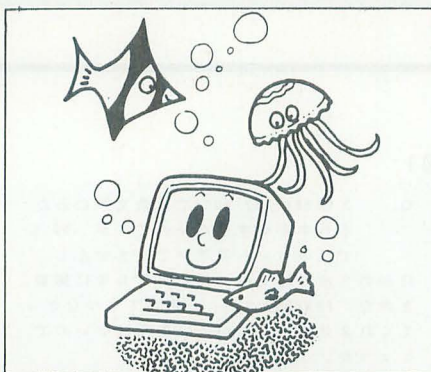
Q　X68000とCZ-89C4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法ではOh!X昨年の7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。
それからWP.Xの環境設定の部分などをいじって文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。
兵庫県　中田　勝啓

図4

Q　X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法ではOh!X昨年の7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。
それからWP.Xの環境設定の部分などをいじって文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。
兵庫県　中田　勝啓

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒102 東京都千代田区
九段南2-3-26井関ビル
㈱日本ソフトバンク出版部
「Oh！X質問箱」係

FROM READERS TO THE EDITOR

この本をみなさんが手にするのは梅雨真っ只中の頃と思います。いま外では雨が降っているでしょうか。雨ってきらいな人が多いと思うんですが，まあまあ好きなんですよね。なんか気分が鬱になりますからね。変人なのかな？　（A）

◆来月（6月号）はフロッピー付きかぁ……。もしFLOAT2＋.XやOPMD.Xが収録されてたらショックだなぁ。だって春休みにアルバイトの合間にちびちびとがんばってうちこんだんだもん……ううっっ。　石間　崇（19）京都府

入っていましたね。かわいそうに……ううっっ。

◆ディスクが付くからって値段が上がるのはコンピュータ雑誌多しといえどOh!Xぐらいじゃないか？　中身がよければ許す！

北条　章（37）東京都

中身はともかく，普通に原稿料を払えばあれくらいにはなるんですよ。

◆SUPER-HDは本屋さんでもらえるのですか。みんなが知ったらきっと本屋さんがつぶれるぐらいの人が来そうだから，予約しといたほうがいいのだろうか。ありがとうございます。

梶田　真二（16）奈良県

だから，ちがうって……。

◆6月号の特集「Oh!Xも8周年だからたった220円ぽっち増えただけでディスクを付けてやるぜ！　ありがたく思え」に期待させていただきます。　小堀　昌宏（21）兵庫県

そんないいかたしなくても……。

◆6月号はすごそうですね。悪魔のツールってなんなのでしょう。気になります。ところで，読者から投稿されたプログラムの入ったディスクはどうなっているんでしょう。ふと気になってしまった。そうとうの数になっていると思うんですが。ま，いいけどさ。

奈良　雅雄（16）栃木県

読者から投稿されたディスクですか。えっと，フォーマットして使っていま……。ウソですよ。

◆こんばんは。いまは6時21分です。編集室の人たちはどこのフロッピーを使ってますか？また，どこのフロッピーがいちばん良いと思いますか。　安元　英彰（21）兵庫県

だから，読者から送られてきたフロッピーディスクをフォーマットして……（みなさん本当にウソですよ～）。

◆がーん！　5月号 P.44 の発売中のソフトに「ギルガメッシュ・ソーサリアン」が。もう，売っていたとは。その翌日タケルへ走った私は再び，ががーん！　まだ売ってねーぢゃんか。メーカーに電話してもらうと「メドが立たない，6月中には……」だと。「ごめんなさいのコーナー」行きだよ，コレは。

味野　真一（24）岡山県

どうも開発の進行が思ったようにいかなかったようで発売されなかったようですね。どうも，ごめんなさい（ゲーム記事担当者）。

◆ポピュラスは受験生の敵だ！　全国の受験生を代表して僕がポピュラスを踏んづけてくしゃくしゃにしてあげます。今すぐ身柄を引き渡してください。　萩野　真一（17）愛知県

とかいいながら，自分がハマってるんでしょう。

◆懺悔。職業欄に「来年は大学生」とか「春には大学生」などと書いていましたが，桜散るどころではなくて桜猛吹雪となり何も見えなくなりました。ハガキに書いたことはぜ～んぶウソとなってしまいました。これらの罪は重く懲役1年，罰金50万円の予備校の刑に処せられました。来春，晴れて出所のあかつきには「合格したよ～」とご挨拶したいと思います。

立花　真哉（19）茨城県

浪人もなかなかいいものですけど，あんまり予備校生活が長くなりすぎて牢名主にならってしまわないようにしましょう。

◆とーと一社会人になりました。思えば学生時代はなんて自由だったのでしょう。そう思って毎朝学生を見るとやっぱしあの頃はよかったなァと思うけど，お金をカポッともらうと社会人はこれだからやめられねー（？）と思います。

山本　裕治（18）滋賀県

まさにそのとおりですよね。

◆とうとう高3になってしまいました。学校が就職校なので働くことになるんです（もちろん進学もできますが)。友達とそのことを話すとやれアセンブラを勉強してソフトハウスに就職するなどとできもしないことを平気でいっています。僕もどうなるかわからないけど，もしどうしようもなくなったらOh!Xのスタッフに入れてもらえませんか。　村松　智行（17）静岡県

そうか，Oh!Xのスタッフってどうしようもなくなった人間の集まりだったのか……。

◆大学行けませんでした。夢はかないませんでしたが私は一労働者となり一定の「金」が手に入る。フフフ，X68000SUPER-HDを注文したいまとなっては6月が待ちどおしいぜ。C compiler PRO-68Kを注文してこよっと。おっと，色はもちろんチタンブラックさ。

笹野　暢彦（19）静岡県

人間どこに幸せが転がっているかわからないものですね。

◆4月から大学生，しかもひとり暮らし。思いっきりX68000をやるぞ！　と思っていたのに，炊事・洗濯 etc. と結構忙しい。それに自分が家計をあずかっていると，ソフトや周辺機器を買うのもお金がもったいないという気持ちになってしまう……。　松永　正弘（20）京都府

う～ん，大学生ではなくすっかりしっかりものの主婦と化していますね。適当にやっていてもなんとかなるものですよ。

◆いま僕は理科の勉強をしています。あっ，先

◀狩野　裕一　宮城県
このバブルン（ボブルン？）は泡を吹きつけてるんでしょうか。吹き飛ばそうとしてるんでしょうか。あるいは割ろうとしてるんでしょうか。

◀丸藤　俊之　神奈川県
丸藤さんは毎月のように載っているので今回は遠慮してもらおうと思ったのですが。う～ん，これだけのものを送って来られると……。カッコイイ。

◀志貴　彩乃　神奈川県
人物の絵はまあ普通ですが，バックのところの感じがなかなかいいですね。これからもどんどん送ってください。

◀岡村　直也　兵庫県
岡村くんはもう1通4コママンガを送ってくれたんですが，両方ともなかなか面白かったですよ。雰囲気が少し広告マンガっぽいですが。

生に見つかってしまいました。先生の右手が僕のえりをつかみました。「うわ」いま先生の説教を受けています。あー，今度は左こぶしが上がりました。「1秒ごとにエキサイティング」もうだめです。みなさんさよーならぁぁぁぁ……「ボカ」。　芝　哲也（16）広島県

みなさん，くれぐれも見つからないように読みましょう。

◆うちの学校の物理科学部員はPC-9801を運べといわれるといやがる。だが，個人で持ってきたX68000を運んでと頼まれると取りあいをして「うおおー，初めてキャリングハンドルにさわったーっ」と喜ぶ。藤田　真人（16）静岡県

X68000PROの場合はどうなるんだろう。

◆月2回になるとうれしい。

今村　雅（17）長野県

月2回になるとしんどい。

◆最近は内容が濃いので全部読むのに2カ月かかる。　武川　博（20）静岡県

ということは2カ月に1度しか買わずに済むということだ。じゃなくて，ちゃんと毎月買ってね。

◆僕のOh!Xが5冊になりました。並べると厚くなってなんかうれしい。

小掠　昌宏（16）三重県

この本が薄いということがいいたいのでしょうか（深読みしすぎかな）。

◆Oh!Xを初めて買いました。こんなに中身の充実している本を読んだのは辞書以来です。これからも読み続けるのでがんばってください。

柴田　知久（17）東京都

辞書のように中身はあるけどつまらないといいたいのだろうか……（ちがうって）。

◆ようやくX68000を購入しました。Oh!Xも毎月買おうと思います。THE COMPUTERも毎月買っています。C MAGAZINEもときどき買っています。Oh!PCはたまに買っています。月刊情報処理試験，先日たまたま買いました。Oh!FMは以前買ってました。プレゼントください。

有山　州一（31）福岡県

まだまだ甘いですね。パソコン・マガジンやBEEP! メガドライブ，それにPC WEEKも買ってくれないと。

◆みっ，見てしまった。ついに生で見てしまった(変なものじゃないぞ)。学校で後ろの席のヤツが，あわわわ……(おどろき)。そいつは手もふれずスプーンを曲げてしまったのだ！　「いますぐにできる？」と聞くとなんのためらいもなしに「できるよ」。たまたま弁当にスプーンを持ってきているヤツがいたので，それでやってもらった。すると手でひねるように90度まで曲げ，あとはポトリと折ってしまった……超能力は存在する。誰にも文句はいわせない，ひぇ〜。

前多　和洋（16）愛知県

う〜ん。キテます，キテます。

◆日経新聞（4/30）によれば，通産省の諮問機関が5月に行う提言の中には透視力やテレパシーなどの超能力の解明があり，テレポーテーションの研究も候補に挙がっているとのこと。これらのものの実現は2020〜2050年頃が目標だそうですが，いったいどんな社会になるのでしょうね。　伊藤　進（34）愛知県

そりゃもう大変ですよ。なにしろ透視力でしょう。でへへ。

◆僕はマドルのイースIII冒険記がとっても大好きです。マドルのセリフはメチャ笑えます。そんな記事（？）を書く西川さんも好きです。ところで，今度Oh!Xに編集者さんたちの写真と簡単なプロフィールを載せてください。

細谷　和治（14）埼玉県

写真を載せてくれというハガキはよく来るんですが，そんなに見たいのならウチに就職すれば見られますよ。

◆GAME OF THE YEARプレゼントで「RPG秘宝館」が当たったんですが，「えっ？　そんなもん応募したかな，ひょっとしてエッチなゲームソフトかしら……」とバックナンバーを確認してしまいました。しかし「秘宝館」って言葉にはあやしげな雰囲気がありますよね。伊豆地方にもいっぱいあるので一度見に来てください。

大野　二郎（23）静岡県

そうか，いっぱいあるんですか……。じゃあ秘宝館のハシゴでもしようかな。

◆ドラクエIVを解く前に最後のボスが読売新聞の日曜版に出ていたのを見てしまった。面白さが半減してしまった。くそー！

今井　勇樹（17）東京都

そうだったのか，見なくてよかった。私はまだ終わっていないのだ。

◆2月に受験のため東京に来てあの秋葉原に行きました。ある店でX1用アメリカントラック（新品）が50円で売られていたので思わず買いました。家に帰ってびっくり，テープ版だったのです（私はデータレコーダを持っていない）。ほかのソフトはディスク版ばっかりだったじゃないかー。　千葉　生樹（18）宮城県

50円じゃあしょうがないような気がしますけど。

◆池袋にある大手安売りショップに信長の野望を買いに行った。なぜかというと，3月号の「CRISIS in Tokyo」と友人の家でやったポピュラスに影響されてシミュレーションがやりたくなったからである。が，そこにX1シリーズのコーナーがなかったのである。店員に聞くとX1というパソコンさえ知らないというありさま。つくづく時代の流れを思い知らされました……うるうる。　北島　謙次（17）東京都

店員がX1を知らなかったというのがいちばん悲しい。まあ，たまたまそうだっただけのことだろうけど。

◆国際無線がまずなくなって，そして千石通商が半分に小さくなって，某量販店が建つことになってしまった。秋葉原ではあの通りがいちばん好きだったのに非常に残念だ。あれじゃ「再開発」の名を借りた「破壊」じゃないか。やっぱりアキバの魅力ってオールマイティじゃなくてアクの強さだと思う。

幅田　圭吾（18）茨城県

私はまだ秋葉原に行ったことないんです。大阪の日本橋なら何度も行ったことありますけど。ああいうのは残してほしいですよね。

◆祭りで買ってきた金魚はどうして長生きしないのか。また1匹……はかないものだ。

小林　宏昭（17）新潟県

僕が幼い頃に買ってもらった金魚は鯉ぐらいの大きさになるまで生きたので，結局学校の池に放してあげました（ひょっとして金魚じゃなかったのかな）。

◆5月5日は僕の誕生日でした。10代最後の19歳となりました。来年はもうおじさんかぁ。

井上　敬介（19）神奈川県

そうか，20歳以上っておじさんなのか。自分がおじさんになっていたなんて全然気がつかなかった。

◆「天国に一番近い」ニューカレドニア島の北東400kmのところにエロマンガ(Eromanga)島という島があるが，そこで日本語学校を開いたら島の住民は島の名前を変えるかもしれない。

松浦　範明（16）広島県

そういえば手塚治虫の「三つ目がとおる」で和登サンがエロマンガという女性と間違われる話があったなぁ。

◆少し前のことだが「Japan Clasic」というアメリカの大学のバスケットボールのオールスター

戦をテレビで見ていたら，ベンチの控え選手がアルファエーを飲んでいた。はるばるアメリカからやって来て試合中にあんなものを飲まされる選手たちは気の毒である。

松井　和宏（21）東京都

実は相手チームの陰謀だったりして。

◆「PC-286対応S-OS“SWORD”（互換機可）」というところがやっぱりOh!Xの良いところ(?)だと思う。　三村　康夫（19）福岡県

いやぁ，そんなにほめられると照れちゃうな。

◆結婚3年目を迎えようとするこの頃。嫁入り道具のひとつとしてやってきたX68000は，仕事の忙しい私の知らないところで妻の手により，すっかりゲーム専用機とされているようです。増えていくソフト，そして平成2年1月4日に長男誕生。X68000が母子により私の手から奪われる日は近い……？　戸崎　篤典（28）群馬県

もともとは奥さんのものなんでしょ。いいじゃないですか。

◆私のX68000ACEは近くの工事のミスのため昇天してしまった。そして再起不能となってしまったので弁償してもらうことになった。ほかの家電のうち直せるものは直したうえでその費用を払ってもらったが，なにしろトランス（電柱についてるゴミバケツみたいなもの）が落ちて高圧電流が我が家をかけめぐったのだから愛機は再起不能となったのである。ところが買い換えようとするとACEは流通在庫のみしかなく，それにEXPERTIIが出るのでそれを待ってはどうかとのことで，最近ようやく手に入れたのであった。　北風　裕介（18）兵庫県

ACEがEXPERTIIになったんだから，やっぱり得したということになるんだろうな。

◆X68000を買うまでず～っと使っていたMZ-80Bを友達に譲ることにした。パソコンについての知識はまったくないそうだが，プログラムを作れるようになりたいといっていた。そこで，こっちからこの話を持ちかけたわけである。MZにしてみればホコリをかぶっているよりも誰かに使ってもらったほうがしあわせですよねぇ？

伊藤　盛人（16）福島県

そりゃそうですよ。ちなみに私もPC-6601SRを友人にあげました。しかし，私自身は現在パソコンを持っていないのでした。

◆4月6日シャープからDMが届いた。いつもの渋い白い封筒ではなく派手な真っ赤な封筒だった。中を開けてみて「えっ，X68000に新機種！」とびっくりしました。まさか仕様は変えていないだろうな，たぶんそんなことはないと思うけど……。そして，おそるおそる仕様表を見ました。「ゲッ，変えてる！」，その仕様表には「BIOSスピードアップ（平均2倍）」と書いてあったのです。シャープに文句言ってやろうかと思いました。だってBIOSってROMでしょ，従来機のパンフレットにもそう書いてあるし。ROM変えたら交換のメンテナンスを受けるしかバージョンアップの方法はない。その日は落ち込んだ状態のまま寝ました。次の日，またシャープからDMが届いた。今度は例の渋い白い封筒だ。内容は春の見・体・験フェアのことだった。招待状も入っていた。よしこれに行って真相を明らかにしてこようと思った。4月14日学校の帰りにNSビルに寄りシャープの人にこのことを聞いてみた。答えは「ROMは一切書き換えていません。デバイスドライバの拡張です」とのことだった。僕にはその声が神様の声に聞こえた。

山口　隆久（17）東京都

このことについてはけっこうハガキが来ていたのですが，そういうことですからみなさん安心してください。まあ，5月号の記事の中にも書いてあったのですが。

◆先日ショットバーでフィンランド製の缶ビールを見ました。そのラベルにはなんと東郷平八郎の写真が印刷されているのです。フィンランドと東郷平八郎の関係について誰か知っている方がいたら教えてください。

山崎　真二（34）埼玉県

というわけで知っている方がいたら教えてあげてください。私は全然知りませんから。

◆1986年2月号よりずっとOh!MZ（Oh!X）を購読しています。連休中にじっくりと読み返してみましたが，参考となる記事の多さにあらためて「この雑誌は良い雑誌だ！」という認識を深めました。私のMZ-2500はメーカーのサポートこそなくなりましたが，3つのモードすべてでシステムソフト（もちろんS-OS含む）が走るのでOh!X（THE SENTINEL）のあるかぎり成長を続けるたのもしいマシンです。3モードそれぞれに個性があって面白いですよ。私もいつかは胸のすくような面白いソフトを投稿したいものです。編集部のみなさん，どうかこれからも魅力ある雑誌作り頑張ってください。それから健康にはくれぐれも気をつけてください。

春日　貴幸夫（32）鳥取県

どうもありがとうございます。そういっていただけるとこちらもすごくやる気が出ます。

◆そうですか，もう1000$_B$年になりますか。早いものですね。霧降高原からのYumiさんはお元気でしょうか。昔のOh!MZを見ていて，ふと目に止まるのはいつもこのページです。あれは1ページだけど全ページの中で唯一難しい話題から離れられた「オアシス」のような存在でした。もしかしたらオアシスそのものだったのかもしれませんね。いま読んでもとても感じのいい文章で好きです。また良かったら再開してほしいですね(パロディも出たことだしね)。お願いします。　屋野　明正（18）大阪府

いまある記事の中でオアシスのような存在というとどれになるんだろうか。やっぱりそういうのは絶対必要ですよね。

◆スタッフになってみたいと思うのだがなれそうにない。5年間パソコンを使っていてもまともにプログラムも組めない（できない度95%）。また，東京近郊とも言い難い。茨城県でもかなり東京寄りなのだが最寄りの駅まで車で30分弱かかるのである。駄目押しは文章力がない。残念である。　倉田　泰幸（20）茨城県

本当に残念だなあ。スタッフになるといろいろいいこともあるのに，ふっふっふっ。

◆5月号のちゃだワに載った畦地君，実は僕も最初の機械はM5だったのですよ。あの頃はよかった。ところであなたのM5どーなってます？私のは完全に引退してベッドの下で眠ってますけど……。私は根っからのゲーマー，そしてホビーストなので間違ってもビジネスマシンであるPC-9801なんぞ買いません。私がパソコンに望むのは正にパーソナルワークステーション，つまり「自由に目的を決め，自由に使うことができるシステム」（取扱説明書より）。やっぱいろいろやってみたいもんねぇ，自分で。でもM5は惜しかった，俺大好きだったのにな。しかしX68000とM5って似ていると思いませんか？「パソコン」のとらえ方が。

林　祐一（19）東京都

やっぱりみんな最初に使ったマシンというのは心に残るんですね。私もやっぱりPC-6001のことは忘れないでしょう。

◆いま思えばノーランズのセクシーミュージックが流行っていたあの頃（ちょっとずれてるかな）。MZ-80Kで「人間やめますか？　それとも……」というほど面白かった。バグ・ファイヤーをやっていたあの頃「なになにシャープ専門誌が出たんだって。女の人の絵が表紙かあ，(裏

◀尾澤　宏　兵庫県
上手なのかどうかはわからないけど，私はこういうタッチの絵は好きです。がんばってX68000を手に入れてください。

◀住友　智代　愛媛県
住友さんはアンケートハガキとイラストのハガキの両方合わせていったい何通送ってくれたんでしょう。がんばって作物作ってください。

返して)ちょっと高いなあ買うのよそっと」(ごめんなさい)という出会いから数えてついに通巻100号ですかへ。おめでとうございます。これからも楽しい記事を載せられるようガンバッテください。　松澤　克明(22)東京都

さあ,いよいよ来月号で通巻100号です。記念になにかパァーッとやりたいですね。

◆X68000のウイルス事件が新聞やテレビで取り上げられていましたが,いまいちわけのわからない報道ばかりでしたね。もう少し報道側もコンピュータについての予備知識くらい持っていてほしいですね。幸いサンデーネットなどにもワクチンがアップされていたので一安心ってところですが,ウイルスを作ること自体をなくせないものでしょうかね。カナシイ……。

佐藤　誠(30)東京都

まあ今度のことで某紙はかなりいろいろといわれているようなので,これからはもっと慎重な対応をするようになるでしょう。

◆X68000の現存数の過半数にウイルス汚染が発生したというニュースを見た。こういうときにこそユーザーあるいは次期ユーザーの方々が団結し,このような問題をはねかえしX68000のポテンシャルと位置づけを高め,より多くの人の支持を得る必要があると思う(逆境を乗り越えてこそ次のパワーを生むものです)。

秋保　三秋(31)埼玉県

過半数がやられてるだって? どこのバカがそんなことをいったんでしょう。実際には1%にも満たないと思いますよ。

◆コンピュータウイルスのせいでX68000の人気が落ちることはないでしょうか?

河野　幸孝(16)山梨県

ウイルスは大問題だけど,みんながしっかりしていればそんなことにはならないと思います。どういうわけかかえって人気が上がってたりするんですよね。

◆1990年4月29日,私はなにげなく1989年5月号の「ちゃだワ」を読んでいた。そして,ふと目に止まった1通のお便り……。千葉県の石崎賢さん! 私も小山田いくさんの大ファンなんです。それも「ぶるうピーター」の頃からの……。クラスの中にはファンの人がひとりもいなく淋しい私。どうか私と文通してください。そして,いく先生の良さを語り合いましょう。〒791-01 愛媛県松山市溝辺町47-1

住友　智代(17)

この方は何通も同じ内容のアンケートハガキを送ってきてくれたので,しょうがないから最後にのせてあげました。どうかお手紙書いてあげてください。

▲寺門　修司　兵庫県
なんかホノボノとしていていいですね。やっぱり夏といえば海です。今年の夏はこういう誰もいない海岸へ行ってみたいな(無理かな)。

ぼくらの掲示板

●掲載ご希望の方は,官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買,交換については,いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合,掲載できない場合もあります。

仲間

★このたびX68000ユーザーのサークルを作ることになりましたので会員を募集します。活動は月1回の会報の発行と情報交換などです。興味のある方は62円切手同封のうえご連絡を。初心者の方も大歓迎です。〒151　東京都渋谷区本町2-29-11コーポ丸山14号　高橋賢一郎(18)

★当方ではX68000ユーザーを対象としたサークルを作るにあたってスタッフor会員を募集しています。内容はディスク会報の発行などをしたいと思いますがそのほかにも会員の意見を取り入れたいと思います。初心者,女性大歓迎ですので,入会希望の方は62円切手同封のうえご連絡を。〒300-04　茨城県稲敷郡美浦村大谷246-1　小沢正人(16)

★「NANNO CLUB PRO-68K」では,X68000ユーザーを対象に南野陽子のファンを募集します。ほかでは手に入らないSX-WINDOW用のアプリケーションを配布します。連絡は62円切手同封のうえ封書で。〒336　埼玉県浦和市常盤3-7-14　武山智裕(18)

売ります

★CZ-855Dブラックを2万7千円,CZ-8BV1を5千円,CZ-8BS1を7千円,CZ-8RL1を3千円で。連絡はハガキで。〒526-01　滋賀県東浅井郡びわ町落合726　大森康雄(23)

★マウスCZ-8NM2を3千円,プリンタケーブルCZ-800Pを3千円。エプソンVP-80K(トラクタ,PCROM付き)ヘッドピン折れのため漢字難ありを1万円(おまけX68000用ケーブル)。サイバーノート用電子手帳ADPを8千円で,すべて送料込み。連絡は往復ハガキで。〒281　千葉県千葉市三角町487-98　久吉一馬(41)

★漢字プリンタ(X1turbo対応)エプソンRP-80F/TⅡKを送料別で1万5千円。新品同様,付属品一式,マニュアル付き。価格相談可。〒579　大阪府東大阪市昭和町1-7　青地信政(36)

★(譲ります)「それ行け! X1」「それ行け! Xファミリー」誌,創刊3号から最終号まで20冊を無料で。ただし送料そちらもちで。いちばん欲しがっている方にさしあげます。必ず往復ハガキにてお願いします。〒104　東京都中央区晴海1-6-15-309　祐成好規(30)

買います

★MZ-1500用RAMファイルMZ-1R18,完動なら可。送料込みの希望価格を書いてハガキで連絡を。価格優先でおひとりに2週間以内に連絡いたします。〒581　大阪府八尾市弓削町1-191　小枝直隆(19)

★X1用漢字ROMのCZ-8BK2またはCZ-8KKを6千円,X1用ディスクドライブCZ-503Fを1万2千円,X1用IOポートCZ-8EPを4千円で。いずれも送料込み。連絡は往復ハガキで。〒652　兵庫県神戸市兵庫区下沢通2-3-4　吉岡延洋(24)

★2400bpsのモデムを2万円で。連絡はメーカー・機種名明記のうえ往復ハガキで。送料はこちらもち。〒300-24　茨城県筑波郡谷和原村細代41-1　宮本健一(19)

★X1で使用可能な24ピンドットインパクトプリンタを3万〜5万円で。連絡は希望価格明記のうえ往復ハガキで。〒090　北海道北見市東陵町57-10高橋下宿内316号室　宮本康司(19)

★X1用FM音源ボードCZ-8BS1+付属品を1万円以内で。連絡は往復ハガキで。〒999-35　山形県西村山郡河北町谷地字12堂16-15　斎藤一郎(29)

バックナンバー

★Oh!MZ1986年3月号を送料込み2千円で買います。切り抜き不可。連絡は往復ハガキで。〒275　千葉県習志野市東習志野2-2-4-307　千田孝之(20)

★Oh!X1987年10月号,1988年3月号および9月号,1989年8月号を。読めれば傷・汚れは可。できれば4冊セットで。送料込み4千円で。連絡は往復ハガキにて。〒244　神奈川県横浜市戸塚区南舞岡2-12-3　野沢幸弘(14)

★Oh!X1988年12月号,1989年5月号を送料込みで各千五百円で。切り抜き不可。連絡は往復ハガキで。〒569　大阪府高槻市天神町2-30-8　有永聡(21)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今回は，5月号の記事に関するレポートです。

●特集の「BASICプログラミング」についてですが，「みんながやってきたのだから，マニュアルを読め」という態度はよくないと思う。筆者はマニュアルを最初から1ページずつ読めと言っているのではなく，「使いたいコマンドがあれば，一度はマニュアルを引いてみなさい」という意味で言っているのだということもわかってはいます。しかし，あの文章を読んだだけでは「やっぱりこの味気ないマニュアルを読むのか……」と思います。自分が苦労してきたのであれば，あとから来る人には楽をさせてあげるのが務めというものだと思います。

湯澤聡（27）X68000, X1turboIII, MZ-2861/2531, PC-6601, MSX, PC-1360K　埼玉県

●特集ですが初心者には内容がちょっと……。やはり初心者に対してプログラミングを教えるためには，基本中の基本を教えるぐらいでないと。あと，ユーザー定義関数は一度使い方がわかるとすごく便利だと思う。友達のユーザーに教えてあげるがなかなかわかってくれない。自分は独学で使えるようになったが結構たいへんである。さらに，知っていれば便利なグローバル変数とローカル変数の違いも説明してほしい。

原田謙（15）X68000PRO　広島県

●「潜入！　バグ対策24時間，X-BASICはいま……！」は非常にわかりやすかった。個別の例をあげてくわしく説明してあり対策方法もわかりやすかった。プログラムを書くのはそれほど苦労しなくてもできるようになると思いますが，問題はそのあとのバグ取りです。これは実際に数多くこなさないとなかなか上達しない（正確なプログラムを書けばバグはでないという話もありますが）。こんなときのためにもバグ取りの記事などは読者にとってプラスではないでしょうか。特に最後のバグ取り最終奥義はおそらく実体験にもとづいたものでは？　自分の体験に重ねて非常に面白く読みました。

高田博（32）X68000ACE-HD, PC-8001 和歌山県

●特集の中でいちばん面白かったのが「プロトタイピングのすすめ」です。いままでプログラムはフローチャートから組まなければスパゲッティになるといわれてました。しかし，この記事にはしろうとのプログラムはある程度大きくなってもプロトタイピングがよいということが書かれていて賛成できました。

また，「ラジコンスティックの製作」ではパソコンとの接続うんぬんよりもラジコンの発信・受信の仕組みがわかったところがたいへん面白く読めました。今後はX68000ユーザーに限らず誰にでもできる回路の製作を期待したいです。

末吉克行（21）X1G, MZ-731, FM-7　兵庫県

●「ラジコンスティックの製作」はよい意味でマニアック過ぎる内容だったと思います。プロポを使ってアナログスティックを製作してしまうとは……。しかし，しかしです。私の住んでいる静岡というところは俗にいうナムコボタン（スイッチのこと）が1個350円もするのです。しかも，フットスイッチとなると……。これはもう6月号から始まる「ハードウェア工作入門」が低価格でできるものをやってくれることを祈るしかありません。

藤田康一（19）X68000PRO　静岡県

●「言わせてくれなくちゃだワ」ですが，毎年ウケ狙いのハガキだけだと思っていたら結構マジメに考えている人もいるんですね（少数ですけど……)。そんな意見もたくさん載せてください。ところで，構成とかはよいのですがバックカラーをなんとかしてほしかった。136ページから137ページに目を移した瞬間，チカチカときて……（私だけでしょうか)。

また，CARD.FNCを使ったカードゲームコンテストなんてよいのではないでしょうか。

藤原博人（26）X1turboZ＋NEW Z-BASIC　鳥取県

●「言わせてくれなくちゃだワ」はいつも楽しく読んでいます。Xユーザーの本音や思想などを知ることができるので，とても面白いと思います。また一緒に発表されている読者の所有機の変化も興味ある点です。いま現在，MZシリーズはどうなってしまったんでしょう……？　半数がX68000のユーザーであるというのはとても凄いことだと思ったりもします。今回は内容によって分類しているようですが，私は以前あったような地域割りのほうが親しみがあってよいように思いました。もっとも投稿者の住所が書いてあるのでこれでも十分なはずなのですが，その地域ごとの考え方の偏りのようなことは地域割りにしたほうが明確に表れて面白いように思います。今回は地域ごとの読者の割合なんかも出てませんでしたね。これは出ていてもあまり意味がないのかもしれませんが。

森川一（24）X68000ACE-HD, X1turboII　北海道

●前回のカードゲームのパターン定義をこのような形で発表してしまうとはまさに「必要は発明の母」といえるでしょう。CARD.FNCのおかげでX68000ユーザーはあまり苦労することなくカードゲームを数多く発表することができるでしょう。135Kバイトを5Kバイト近くまで（まあデータサイズをですが)，ごくろうさまでした。たしかにカードゲームではパターンは同じものを使用してよいわけですし，それがきれいな絵柄であるならばとても喜ばしいことであります。スートのことなどうんちくも面白くためになりました。BASIC特集にこれを早速活用したNinety-Nineが出ていたのもよかった。

大津和之（20）X1turboZ　福岡県

ごめんなさいのコーナー

6月号　付録ディスクの解凍

P.49　まとめて解凍する方法で，残ったSMPLを手動で展開してくださいというところの手順に間違いがありました。手順の最後の行は，

B>A：LH －E A：SMPL　です。

また，HumanのシステムがVer.1の場合は，おまけディスクをまとめて解凍することはできません。申し訳ありませんが個別に解凍してください。お詫びして訂正いたします。

そのほかはP.130からのアフターケア参照。

6月号表紙絵の作者名が違っていました。作者は須藤牧人さんです。お詫びして訂正いたします。

バグに関するお問い合わせは
☎03(230)7683(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

Oh!X編集部は7月16日より新社屋に移転

▼今月の特集は久びさにマシン語特集となりました(1年半ぶり)。68000を中心としたマシン語入門ではZ80を中心としたマシン語入門とまた違った様相を呈していますね。今回はできるだけ初心者にも読みやすいようコンピュータ入門の立場からマシン語を紹介しようと，懐かしいぜんまいちゃんにも登場してもらいました。いかがでしたか。

また，8ビットのほうでは，ついにリロケータブルアセンブラWZDが完成，ZEDAやREDAで飽きたらなかったS-OSユーザーの皆さんにも活用していただきたいと思います。

▼さて，いよいよ愛読者年間モニタを務めていただく方々のお名前を発表いたします。

浅野　憲(大阪府)，泉　昭彦(東京都)，奥村光雄(埼玉県)，織田　聡(岐阜県)，高橋　毅(埼玉県)，高村　信(東京都)，谷口　正和(石川県)，段　宏太郎(兵庫県)，土谷　興正(兵庫県)，中川　比呂志(東京都)，長谷川　敦士(山形県)，畑　剛志(北海道)，松井　伸康(東京都)，横山　賀一(東京都)，梅本　英之(奈良県)

以上15名（敬称略）の皆さんです。さっそくですが今月号からモニタレポート用紙をお送りいたしますので1年間よろしくお願いいたします。6月号までモニタを務めていただいた皆さんには心からお礼を申し上げます。

▼7月1日から株式会社日本ソフトバンクは「ソフトバンク株式会社」と社名を変更いたします。また，社屋も移転することとなり，これに伴いOh！X編集部の連絡先も7月16日から以下のとおりに変わります。

新住所　〒108 東京都港区高輪2-19-13
NS高輪ビル

☎03(5488)1309（Oh！X編集部）
☎03(5488)1311（バグに関する電話）
☎03(5488)1360（出版営業部）
☎03(5488)1365（広告営業部）

どうぞよろしくお願いいたします。

▼今月のマシン語プログラミング入門は筆者の村田敏幸氏急病のためお休みです。

投稿応募要領

- ●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- ●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。
- ●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル
日本ソフトバンク出版部
Oh!X「㋢㋮㋱」係

S H I F T · B R E A K

▶ジャッキー・チェンの番組を見ていた。BGMに聞き覚えがあると思ったら，三国志IIではないか。うーん，テレビ番組にはピッタリだ。してみると，BGMとゲームミュージックは違うんだなぁ。あ，向谷氏はFM音源のエキスパートでした。カシオペアファンの皆様，ごめんなさい。P.S.緑水Softwareの超高速の対応に感謝します。(H.U.)

▶神様，パンクしてしまいそうです。／もう一刻の猶予もありません。／神様，パンクしてしまいそうです。／サブ辞書が大きくなりすぎて，メモリからはみ出してしまいそうです。／この文章を打ちながら，何時(イツ)くるかわからないエラーが怖いのです。／神様パンクしてしまうそうです。／はやく増設メモリを私の手に……。from君のソバで逢おう。(S.K.)

▶最近，統計分析をするんで某富士通の大型機を使っているのだが，パソコンとはずいぶん違った雰囲気なので苦労するのである。しかも，パソコンでできるようなことを，わざわざ大型でやるというのは非効率的だ。それもこれも，X68000じゃメジャーな表計算ソフトがあまりないからだ。ロータス1-2-3が欲しい……。(亀)

▶ゲーム基板についての情報を兵庫県の末光さんや神奈川県の野沢さんほかからいただきました。どうもありがとう。先日私はナムコシステムIIを購入しました。しかし，かねてから腰を据えて遊びたかった「ワルキューレの伝説」がマニアたちの買い占めにあい，手に入りません。誰か売っているお店を知りませんかね。(善)

▶なくなってみて初めてその存在の大きさがわかるものがある。水に空気にマウスのボタン。健在なときは誰もその大切さに気づかない。先日，バイクのチェンジレバーをふとしたはずみで折ってしまい，2日ばかり不便した。ギアチェンジができないのはつらい。たかだか大学→編集室間のなんと長く感じられたことよ(なら素直に電車使えよ)。(A.T.)

▶エラーがでた。エラーがでた。アセンブルエラーがでた。アセンブルエラーがでた。リンクエラーがでた。リンクエラーがでた。バスエラーがでた。バスエラーがでた。アドレスエラーがでた。アドレスエラーがでた。おかしな命令を実行しましたエラーもでた。おかしな命令を実行しましたエラーもでた。こんなにエラーがでるのは，かっこいい。(Mu)

▶ただし，とても恥ずかしいのだが，このさい連載も終わってしまったので書いてしまおう。実は私，マガジンに連載されていた「キラキラ」の愛読者だったのである。いい作品とはいえないが，作者（および主人公）の“屈折の仕方”に妙に共感を覚えるのだ。あーゆー屈折した漫画って少ないでしょ。若いバンドにはいくつかあるけどね。(K)

▶またも薬師丸ひろ子の映画ははずれだった。しかし，斉藤由貴の映画は当たりが多い。「優駿」はともかく，最近の「君は僕を好きになる」と「香港パラダイス」はどちらも傑作だ。特に「香港パラダイス」は喜劇ながらラストシーンでは思わず涙した。涙もろいのは単に年をとったからではないと思う。

(Field of Dreamsはつまらなかった KO)

▶編集部の雰囲気にもなじみ，やっと一息ついたと思っていたところ，バグ電話をとるようにといわれてしまった。前にも書いたように，ぼくはほとんど初心者だから困ってしまう。特に先月はディスクが付いたから山のように電話がかかってくる。初歩的なものは答えられるようになったけど，難しい内容のものはちょっと……。ごめんなさい。(A)

▶自分でいうのもなんだが，私は衣装持ちだ。“ロック少女風”“お嬢さま風”などその日の気分で着分けている。そんな私をほかの部署の人間いわく「あれは“衣装”じゃなくて“仮装”っていうの」。ふうん，そっか，私は毎日仮装して会社に来ているのか。じゃ，せっかくだからゴジラのぬいぐるみでも……。ソフトバンクっていい会社だなあ。(E.O.)

▶パスワードを忘れる。盾の名前を片っ端から入れる。すぐに飽きる。Paint Mapを選ぶ。敵のパラメータをvery fastにする。敵の大陸を地震で均す。気候を砂漠に変えて敵の民を20人増やしてみる。……勝てない。増やすのを10人にする。……また負けた。しかし，人間の対戦を見てしまうとCONQUESTモードのなんと甘いことか。(U)

▶本誌の編集者は伝統的に，男子は西日本，女子は関東出身者で構成されている。というわけで，先月はウイルスのせいでろくに紹介できなかったA君だが，これがしゃべるしゃべる。以前かまた氏がいっていた関西弁強化パックをつけているのではと思うほど。私なんかせっかく覚えたはずの東京言葉がわやくちゃになっちゃった……？？？(T)

microOdyssey

コンピュータというものに出会ったのはいつのことだったろう。おそらく，中学生の頃に友人からコンピュータの雑誌を見せてもらったのが最初だと思う。当時，ぼくは何人かの仲間とデパートの屋上で「トランキライザーガン」などのアーケードゲームに興じていた。だから，そのとき，「これを入力すればただでゲームができるのか」と思った。

数日後，ぼくたちは違うデパートにマイコン売り場があることに気づいた。遊び場はそこへと変わった。売り場といってもそのころはマイコンを買う人はあまりいなかったので展示場という感じだ。マイコンを一般の人にも知ってもらうということが主たる目的なのだ。

そこにはいろいろなマイコンがずらりと並んでいて，「どうぞご自由におさわりください」とでもいわんばかり。MZ-80K，ベーシックマスターなどがあり，ぼくたちはそこへ雑誌を持ち込んでプログラムを打ち込むようになった。

閉店時間までが勝負だった。あるのは本体のみでセーブはできないから。初心者ばかりだからいろいろなトラブルが起こり，時間はどんどん消費される。エラーが発生するのは当たり前。マシン語のリストが出てくると「なんだ，この数字の羅列は。どうやって打ち込むんだ」。こういうことの連続で，なかなかプログラムを走らせるところまでいかない。

しかし，ある日全部のプログラムを入れ終えることができた。そのプログラムはカマキリ星人が暴れ回るというようなゲームだったと思う。ぼくたちは焦った。すでに閉店直前だったのだ。まだ，マシン語部分のチェックサムの確認が済んでいなかったが，ぼくはたぶん間違えていないと思ったので「とりあえず走らせてみよう」と言った。なかには反対するものもいたが，やはり画面を見てみたいという欲求には勝てずRUNさせることになった。しかし，ぼくたちが見ることができたのは止まったままのカマキリ星人だけ……，ひとりがつぶやいた。「そういえばマシン語って暴走するんだよね」。

そんなこんなでぼくの趣味の項目にマイコンというものが加わった。高校の入学祝いにプログラム電卓「fx-602P」を買ってもらい，そのあとはPC-6001，PC-6601SRと所有機種を変え，Z80のマシン語も使えるようになった。でも，それからしばらくして情熱は薄れてしまった。いろいろな理由があったと思う。

マイコンと距離を置いたまま何年かの月日は流れたが，大学卒業後ソフトバンクに入社した。やはりコンピュータとのつながりは切れずに残っていたらしい。いや，未練があったというほうが正しいかもしれない。さらに，Oh！Xに配属されバックナンバーを読んでいると，こんなぼくを再びマイコンの世界に（趣味として）引きずり込むような記事を発見した。「S-OS」とMZ-700版「ゼビウス」ぜある。

はっきりいって感動した。自分はなんてあきらめが早かったんだろうと反省もした。PC-6601SRは友人にあげてしまったのだが，もっともっとつかいこめばよかった。そして，いまぼくはパソコンを買おうと思い，何にするか悩んでいる。マックにしようかな，Amigaにしようかな，と……。だって，一番好きなX68000はすでに会社の机の上にあるんだもん。　（A）

1990年8月号7月18日（水）発売

特集　ADVANCED 2D GRAPHICS

Oh！X通巻100号記念特別企画

通巻100号記念特大プレゼント

特別企画　実況対戦ポピュラス　祝一平vs西川善司

全機種共通システム　WLK

X68000用カードゲーム　HEART

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(233)3312
	〃	書泉ブックマートBI	03(294)0011
	〃	書泉グランデ5F	03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(257)2660
	八重洲	八重洲ブックセンター3F	03(281)1811
	新宿	紀伊国屋書店本店	03(354)0131
	高田馬場	未来堂書店	03(200)9185
	渋谷	大盛堂書店	03(463)0511
	池袋	リブロ池袋店	03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコン∑上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh！Xの定期購読をご希望の方は，とじ込みの振替用紙の「申込書」欄に何年何月号からをご記入のうえ，**年間購読料6,720円（税込）**を添えてお申し込みください。その際，裏面の通信欄に「○年○月号よりOh!X定期購読希望」と忘れずに明記してください。なお，すでに定期購読をご利用いただいている方には，購読期限終了と同時にご通知申し上げますので，同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

7月号

■1990年7月1日発行　定価560円（本体544円）
■発行人　孫　正義
■編集人　橋本五郎
■発売元　（株）日本ソフトバンク
■出版事業部　〒102 東京都千代田区九段南2-3-26　井関ビル
Oh！X編集部　☎03(230)7681
出版営業部　☎03(230)7670 FAX 03(262)8397
広告センター　☎03(297)0181
■印　刷　凸版印刷株式会社

POWERFUL MEGA-MAGAZINE

BEEP! MEGADRIVE

ビープ！メガドライブ

7月号 定価480円(税込) 好評発売中

総額 **200**万円相当 ブランズウィック ゲームアイデア大募集！

特集

ゲームセンター改造計画

そろそろ「うるさい、雰囲気がよくない、オタクのたまり場」のゲームセンターにさよならしよう！

ゲーム業界リクルート大作戦

もうすぐリクルートの夏がやって来る。
ゲーム業界志望の諸君、準備は大丈夫？

熱血メガドライブ宣言

ナムコ、金子製作所

徹底マスター

サンダーフォースIII フェリオス 大旋風 ムーンウォーカー バットマン

■特別付録
BEEP！メガドライブ
HALF YEARカレンダー

ゲームボーイ専門誌 パワーアップした第2弾だ！

ゲームボーイLIFE VOL.2

定価380円(税込)

- 54本のソフトを総ガイド
- 輝け！第1回ゲームボーイ大賞
- 試験にでないゲームボーイ講座

業界初の完全攻略 オールソーサリアンシリーズ

FALCOM MAGAZINE ［ファルコム・マガジン］

予価630円

- オールアバウト・ソーサリアン パソコン版ソーサリアンの総ガイドに加え、メガドライブ版ソーサリアンも紹介
- オールファルコム・ベスト10 ファルコムユーザー100人が選んだファルコムなんでもベスト10

日本ソフトバンク

展示図書一覧

定価は本体価格です。

書名	定価
MS-DOSいたれりつくせり本	●1800円
プレイMS-DOS	●1900円
UNIX System V プログラマ・ガイド	●12000円
UNIX System V ユーザ・ガイド	●9800円
UNIXオペレーティングガイド	●3000円
OS/2 APIブックⅠ	●2709円
C言語の活用理解	●2000円
C言語の基礎知識	●2500円
C言語の応用50例	●2300円
上級・C言語の応用例50例	●2400円
Cプリプロセッサ・パワー	●2200円
Play the C 上・下	●各1500円
Turbo C入門	●2600円
C++プログラミング	●2600円
Quick Cプログラミング	●2602円
詳説C言語	●4369円
8086アセンブリ言語	●2800円
8086マクロプログラミング	●2600円
Final Ver.4.0ブック	●2400円
MIFES Ver.4.0ブック	●2400円
ビジネスソフトデータ活用ブック	●2800円
BASICによるプログラミングスタイルブック	●1800円
ソーティング・ノート	●1900円
J-3100パワーユーザーブック	●2400円
続・PC工作入門	●1800円
PC-286Lブック	●1700円
試験に出るX1	●2800円
RDBファラオ活用ガイド	●2903円
言図ガイド	●2301円
Rydeenガイド	●2427円
P1EXEガイド	●2524円
Lotus1-2-3ガイドⅡ	●2500円
MS-Chart Ver.3.1ガイド	●2900円
まいと～くガイド	●2300円
新松ガイド	●2000円
一太郎Ver.3ガイド	●2500円
新一太郎ガイド	●2300円
桐Ver.2ガイド	●2500円
花子応用ガイド	●2500円
Lotus1-2-3ガイド	●2400円
P1ガイド	●2300円
Ninja2ガイド	●2300円
Multiplan Ver.3.1ガイド	●2400円
アセンブラCASL入門	●2000円
ハードウェア徹底マスター	●2500円
FORTRAN徹底マスター	●2800円
情報処理の基礎知識	●1600円
COBOL徹底マスター	●2900円
受験用語ハンドブック	●1800円
情報処理入門1・2	●各1204円
CASLで学ぶアセンブラ言語入門	●2204円
バイト&ワードの風にのって	●1800円
田原総一朗のパソコンウォーズ	●1400円
パソコンを襲う知的独占の戦い	●1600円
RPG幻想事典・日本編	●1800円
魔法王国シムルグント	●1800円

地域	書店	電話
国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈甲信越・北陸〉		
甲府市	文教堂甲府店	0552-22-4600
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
上田市	平安堂上田店	0268-22-4545
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
上越市	パソトピア 'コスモス	0255-25-5867
山北町	BOOKメディア	0254-77-3850
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店ＳＢＳ店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
〃	丸善名古屋支店	052-261-2251
〃	丸善ブックメイツセントラルパーク	052-971-1231
〃	日進堂上前津店	052-263-0550
名古屋市	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
〃	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
豊川市	三洋堂豊川店	05338-3-0334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイパス店	0584-74-7766
一宮市	三洋堂一宮店	0586-77-5734
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店11ビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
〃	オーム社書店竹田店	075-644-2611
奈良市	駸々堂大丸店	0742-26-6241
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	ヒバリヤ書店ナンバ店	06-644-5407
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
福山市	ブックシティ啓文社	0849-25-0050
〃	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
〃	文栄堂	0839-22-5611
下関市	中野書店	0832-22-6181
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東口店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	積文館新天町店	092-781-2991
〃	福岡金文堂本店	092-741-2106
〃	福岡金文堂朝日ビル店	092-431-1094
〃	福岡金文堂デイトス店	092-451-6175
〃	福岡金文堂アニマート原	092-844-0088
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルバーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-2183
宮崎市	中央、田中書店	0985-24-3511
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館佐賀店	0952-24-4314
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂書店	0956-22-4214
熊本市	紀伊國屋書店熊本店	096-322-5531
〃	長崎書店	096-353-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

HOST

X68000専用
多回線 ホストソフト

PRO-68K

ついに登場!

3回線/9回線

きみも、今日から局長さん

HOST 9 PRO-68K 概要

対応回線数	1〜9回線
使用モデム	ATモデム MNP(RTS/CTS)可
通信速度	最大9600bqs
会員数	*最大9999人
掲示板数	*最大40個
機能	電子掲示板・電子手数・電子会議(チャット)・会員情報

これらは、コンフィグファイルで設定できます。

注1:*印について拡張を希望する場合は、プログラムの書き換えが必要になりますので、別料金にて対応致します。当社までご相談ください。
2:2回線以上で運用される場合は、CZ-6BF1(シャープ純正)が必要になります。
3:このホストはテキスト形式の転送方法を採用しております。

■特長

●各種設定のコンフィグファイル化。●RS-232C回線とは別にキーボードからのアクセス、ダウンロード、アップロードが可能。●モニタで、各チャンネルのユーザーの打ち込んだコマンドや通信状態を確認。●各掲示板別にSIG、ボードパスの設定。●メンテナンス作業のオンライン実行。(ボードインデックス,メールインデックス)●オンラインサインアップ等、ゲストへの設定が可能。●通信サービスTri-P対応。●行編集(オンライン簡易エディタ)機能。●その他・シスオペレベルで会員情報の変更が可能。タイムアウトによる回線切断。PDS専用掲示板の採用。(1書込中で、ドキュメントとテキストプログラムの分離)。●接続MNPタイプの識別。●ログイン、ログアウト時間の記録。●非アクセス時のモニタ画面消去可能。

HOST 3 PRO-68K

機能は統べて、「HOST 9 PRO-68K」と同じですが、対応回線数が、1〜3回線に制限されて、低価格でユーザーに供給します。

バージョンアップ (Ver1.10) サービス実施中

現在発売されています製品は、Ver1.10に変更になっています。お使いの製品がVer1.00のユーザーの方のために、バージョンアップサービスを実施しておりますので、お早目に、ユーザー登録葉書をお送り下さい。
Ver1.10へ無料交換を実施しております。

好評発売中

HOST 9 PRO-68K ¥59,800円
HOST 3 PRO-68K ¥39,800円

SPS-NET
TSUKUMO-NET モデル運用中!!

たーみのる 2

今、X68000の通信が変わる!!!

ユーザー重視の機能を搭載して
好評発売中
17,800円

24/31KHz ディスプレイ対応

「た〜みのる」が
装いも新たに
「た〜みのる2」として登場!
「た〜みのる」が
通信人門版なら
「た〜みのる2」は
マニアタイプの
通信ソフトです!!!

X68000専用
パソコン通信ソフト

「た〜みのる2」はX68000用に製作された通信ソフトです。X68000の機能を充分に引き出して、ユーザーの方々が簡単に操作できるよう工夫・製作されています。

X68000 HOST PRO-68K 使用 SPS-NET

TEL (0245)46-1167(代)

Tri-P 好評運営中(4回線)/一般回線(9回線) MNPクラス7

24時間運営(N81XN)
ゲストID(GUEST)
※GUESTアクセスは無料ですのでぜひ、一度試してください。

入会方法

登録料¥3,000(税別)
会費無料

下記の用紙に直接記入するか又は、コピーして記入し、72円切手同封の上、「SPS-NET係」までお送り下さい。届き次第、仮登録を行いID発行後SPS-NET専用の郵便振込み用紙ならびに運用の手引きをお送りいたします。それに従い、3ヶ月以内に登録料3,000円(税別)を御入金下さい。
入金確認後正式会員として再登録します。

例◎パスワード=SPS-NET (8文字まで大小文字の識別あり)
( )
◎本名=大和大五郎(8文字まで)
( )
◎ペンネーム=大ちゃん(4文字まで)
( )
◎年齢=30(現在の年齢)
( )
◎電話=0245-45-5777(市外局番から)
( )
◎職業=株式会社エス・ピー・エス(16文字まで)
( )
◎住所=福島市太平寺字町ノ内5-3(24文字まで)
(〒)
◎自己紹介=SPS-NETをよろしく (24文字まで)
( )
◎システム構成=X68000ACE-HD MD2400B (18文字まで)
( )

(株)マイコンハウス
SPS

〒960 福島市太平寺字町ノ内5-3 ☎(0245)45-5777
FAX(0245)45-1804(GII,GIII)

■表示価格に消費税は含まれておりません。

当社の製品は全国の有名デパート、パソコンショップでお求めになれます。尚、お求めになれない場合、郵便局にてお申し込みください。●口座番号 郡山5-12298 ●加入者名(株)エス・ピー・エス ●金額 代金に3%の消費税を加算した額●通信欄(裏面)ご希望ゲームソフト名、数量、代金合計、年齢、氏名、機種名、テープかディスクの種類。(一週間以上かかりますので、お急ぎの方は現金書留をご利用ください。その場合、おつりのいらないようにお願いします。

信頼と実績のお店
BASIC HOUSE

X68000を御買上げの方にもれなく
下記X68000グッズのいずれか1つを
プレゼント！

A. PROSTAFF ジャンパー
B. X68000目覚し時計
C. ツタンカーメンZIPPO
D. ビジネスバッグ

PRO SHOP & STAFF X68000

サポート万全！ 我々にお任せください！
ボーナス一括払・長期クレジットOKです!!

NEW X68000 SUPER-HD

○ 大容量80MB 3.5″HD内蔵
○ SCSIインターフェイス標準装備
○ 疑似マルチタスク、マルチウィンドウを実現した"SX WINDOW"を搭載
○ 処理速度大幅向上（平均2倍）

BASIC HOUSE 超特価
（限定品）

NEW X68000 EXPERT II

NEW X68000 PRO II

CZ-663C
CZ-605D
CZ-8PC3
定価¥609,800
BASIC HOUSE特価

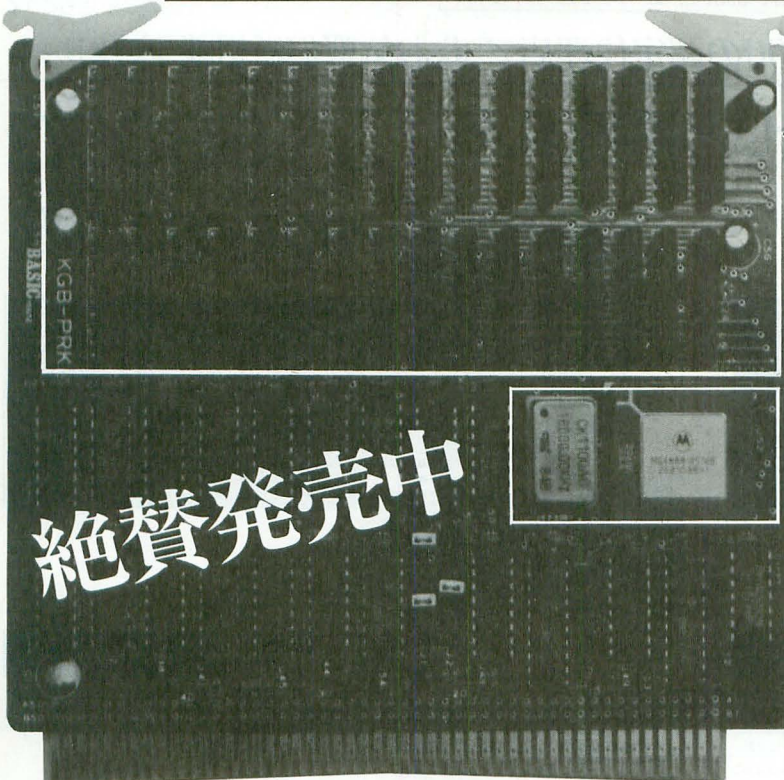
品名	価格	品名	価格	品名	価格
CZ-8PC3	BH特価	HXD042	¥128,000	SX-68M	¥19,800
CZ-8PC4	¥99,800	IT-X640	¥158,000	C Compiler PRO-68k	¥39,800
CR-3415CL	¥109,800	IT-X680	¥188,000	Mu-1	¥19,800
CR-3410CL	¥98,000	MD24FS5	BH特価	マジックパレット	¥19,800
VP-2050	BH特価	MD12FS	BH特価	Zs STAFF PRO-68k	¥58,000
CZ-8NS1	¥188,000	XE-1 PRO	BH特価	C-TRACE68	¥68,000
GT-6000	¥178,000	CYBER STICK	¥23,800	CARD PRO68k	¥29,800
GT-4000	¥198,000	CZ-6BF1A	¥38,000	CZ-6EB1	¥88,000
HS-10R II	¥49,800	CZ-6BG1	¥59,800	CZ-8NT1	¥13,800
HXD040	¥118,000	CZ-6BM1	¥26,800	AN-S100	¥36,600

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部／マイコンショップ／通販部　宇都宮市竹林町503—1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所／マイコンショップ　大田原市美原1—13—4　TEL0287-23-5352　FAX0286-23-5364
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811（代）

2枚のボードが1枚になった

KGB-X68PRK

※写真はKGB-X68PRK-14です

広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
数値演算プロセッサ

- メモリアクセスノーウェイトによる高速アクセス
- CZ-6BE2/CZ-6BE4/CZ-6BP1との混在が可能です
- 複数枚のKGB-X68PRKの実装が可能です
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能です
- ジャンパの変更により数値演算プロセッサの1枚目2枚目/未使用の選択が可能です
- 1M/2M/3Mメモリモデルは購入後にメモリをボード上に追加可能です
- 数値演算プロセッサにはデバイスドライバ(FLOAT3X)が付属します

※CZ-602C/CZ-612C以外の機種ではCZ-6BE1/CZ-6BE1Aを実装している必要があります
※メモリアクセスノーウェイトのため拡張I/O BOXでは動作しません

製品価格一覧

製品	価格	製品	価格
KGB-X68PRK-01 (1Mメモリ/数値演算プロセッサ無し)	¥ 58,000	KGB-X68PRK-11 (1Mメモリ/数値演算プロセッサ付き)	¥ 96,000
KGB-X68PRK-02 (2Mメモリ/数値演算プロセッサ無し)	¥ 74,000	KGB-X68PRK-12 (2Mメモリ/数値演算プロセッサ付き)	¥ 112,000
KGB-X68PRK-03 (3Mメモリ/数値演算プロセッサ無し)	¥ 98,000	KGB-X68PRK-13 (3Mメモリ/数値演算プロセッサ付き)	¥136,000
KGB-X68PRK-04 (4Mメモリ/数値演算プロセッサ無し)	¥122,000	KGB-X68PRK-14 (4Mメモリ/数値演算プロセッサ付き)	¥160,000

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ MC68881RC16	¥38,000

充実のBASIC HOUSEハードウェア&ソフトウェア

製品	価格	製品	価格
高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000	高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000	汎用ローコストA/D&PIOボード(KGB-X1S) X1	¥ 19,800
ハードディスクインターフェースボード(KGB-HDIF) X1	¥ 16,000	高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO)X68000	¥ 68,000	64180CPUボードMach180(KGB-CPXB) X68000	¥ 98,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥ 24,800	ローコストMIDIインターフェース(MELODY BOX) X68000	¥ 16,800

製品	価格	製品	価格	製品	価格
BASIC拡張関数パッケージ(B6-6301)	¥9,800	C言語ライブラリ(B6-6305)	¥6,800	BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	¥ 14,800
ディスクキャッシャー(B6-6304)	¥6,800	Toys & Tools (B6-6307)	¥6,800	アイコンエディタ(B6 -6303) ¥4,800 / CP/M68Kエミュレータ(B6-6302)	¥ 19,800

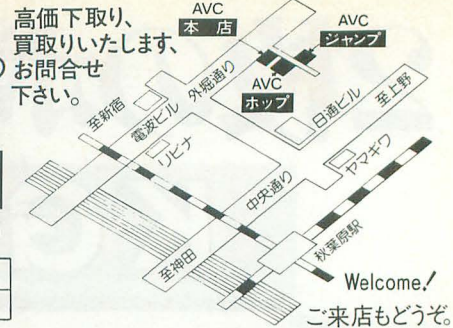
高解像度モードRGB出力をシャープ液晶ビジョンで投影! Macintosh, X68000, PC-9801対応液晶ビジョンインターフェース間もなく登場!

BASICHOUSE BBS TECOSYS NET TEL 0286-27-1829 /1200/2400ボーMNPクラス5/8ビット/パリティ無し/X制御無し ゲストIDなし(オンラインサインアップ)

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部/マイコンショップ/通販部　宇都宮市竹林町503-1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所/マイコンショップ　大田原市美原1-13-4　TEL0287-23-5352　FAX0286-23-5364
マイコンショップ BASIC HOUSE
お申し込み・お問い合せは ☎0286-22-9811(代)

AVCフタバ

〒101 東京都千代田区外神田3-2-3 ☎03-253-7611(代)

今すぐ もよりの電話から	仙 台 022-264-3704	名古屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

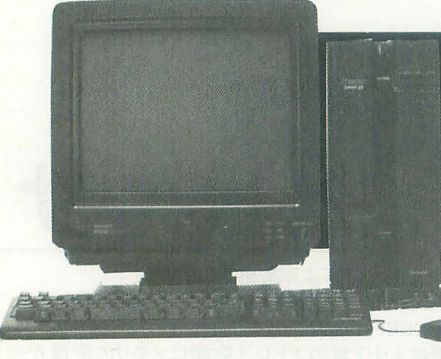
高価下取り、買取りいたします、お問合せ下さい。

AVC本店 AVCジャンプ AVCホップ 至新宿 電波ビル 外堀通り リバナ 日通ビル 至上野 ヤマギワ 中央通り 至神田 秋葉原駅

Welcome!
ご来店もどうぞ。

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

X68000 待望の新しい仲間登場!!

X68000 PERSONAL WORKSTATION EXPERTII・EXPERTII HD

EXPERTII・EXPERTII HD
集積度を高めた"マンハッタンシェイプ"3Mの大容量メモリを搭載。本格的なウインドウシステム、SX-WINDOW搭載。

〔写真のモニタは別売です。〕

CZ-603C 標準価格¥338,000
CZ-613C 標準価格¥448,000
AVC特価

X68000 PERSONAL WORKSTATION PROII・PROII HD

PROII・PROII HD
拡張I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。
もちろん、SX-WINDOW搭載。

CZ-653C 標準価格¥285,000
CZ-663C 標準価格¥395,000
AVC特価

X68000

在庫稀少価格はお電話で!
CZ-602C
CZ-612C
CZ-652C
CZ-662C

お勧めディスプレイコーナー 組合せは自由、価格はお気軽にご相談下さい。

CZ-604D 標準価格¥94,800 AVC特価
●0.31mmドットピッチ ●2モードオートスキャン ●ステレオスピーカ搭載 ●チルト台同梱

CZ-613D 標準価格¥135,000 AVC特価
●ドットピッチ0.31mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱

CZ-603D 標準価格¥84,800 AVC特価
●0.31mmドットピッチ ●TVチューナ無し ●3モードオートスキャン ●チルト台同梱

CU-21HD 標準価格¥148,000 AVC特価
●0.52mmドットピッチ ●21型ディスプレイ ●3モードオートスキャン ●ステレオスピーカ搭載

CZ-605D 標準価格¥115,000 AVC特価
●ドットピッチ0.39mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱

CZ-602D 標準価格¥99,800 AVC特価
●ドットピッチ0.39mm ●TVチューナー搭載 ●チルト台同梱

型番	品名	標準価格	販売価格
CZ-6TU	システムチューナー	¥33,100	AVCフタバ特価
BF-68PRO	CRTフィルター	¥19,800	AVCフタバ特価
CZ-8NS1	カラースキャナー	¥188,000	AVCフタバ特価
CZ-6BN1	スキャナー用パラレルボード	¥29,800	AVCフタバ特価
CZ-6VT1	カラーイメージユニット	¥69,800	AVCフタバ特価
CZ-8BV2	カラーイメージボード	¥39,800	AVCフタバ特価
CZ-8BR1	立体映像セット	¥29,800	AVCフタバ特価
CZ-8DT2	パーソナルテロッパ	¥44,800	AVCフタバ特価
CZ-8BS1	FM音源ボード	¥23,800	AVCフタバ特価
CZ-8NJ1	ジョイカード	¥1,700	AVCフタバ特価
CZ-8NM2A	マウス	¥6,800	AVCフタバ特価
CZ-8NM3	マウス・トラックボール	¥9,800	AVCフタバ特価
CZ-6SD1	システムラック	¥44,800	AVCフタバ特価
AN-S100	アンプ内蔵スピーカー	¥36,600	AVCフタバ特価
CZ-6EB1	拡張I/Oボックス	¥88,000	AVCフタバ特価

型番	品名	標準価格	販売価格
CZ-8PG1	24ピンカラープリンター(80桁)	¥130,000	AVCフタバ特価
CZ-8PK10	24ピンプリンター(136桁)	¥97,800	AVCフタバ特価
IO-735X	カラージェットプリンター	¥248,000	AVCフタバ特価
CZ-6BE1A	1M増設RAMボード	¥38,000	AVCフタバ特価
CZ-6BE2	2M増設RAMボード	¥79,800	AVCフタバ特価
CZ-6BE4	4M増設RAMボード	¥138,000	AVCフタバ特価
CZ-6BP1	数値演算プロセッサー	¥79,800	AVCフタバ特価
CZ-6BC1	FAXボード	¥79,800	AVCフタバ特価
CZ-6BM1	MIDIボード	¥26,800	AVCフタバ特価
CZ-6BU1	I/Oボード	¥39,800	AVCフタバ特価
CZ-6BL1	LANボード	¥268,000	AVCフタバ特価
CZ-243BS	サイバーノート	¥19,800	AVCフタバ特価
CZ-240BS	スティショナリー	¥14,800	AVCフタバ特価
CZ-223CS	通信ソフト	¥19,800	AVCフタバ特価
ゲームソフト		20% OFF	

型番	品名	標準価格	販売価格
CZ-8TM2	モデムユニット	¥49,800	AVCフタバ特価
CZ-252MS	Musicstudio	¥28,800	AVCフタバ特価
CZ-247MS	MUSIC(MID)	¥28,800	AVCフタバ特価
CZ-221HS	NEW Print Shop	¥19,800	AVCフタバ特価
CZ-228BS	TOP給与計算エキスパート	¥200,000	AVCフタバ特価
CZ-227BS	TOP財務会計	¥200,000	AVCフタバ特価
CZ-220BS	DATA	¥58,000	AVCフタバ特価
CZ-212BS	BUSINESS	¥68,000	AVCフタバ特価
CZ-219SS	OS-9	¥29,800	AVCフタバ特価
CZ-211LS	Ccompiler	¥39,800	AVCフタバ特価
CZ-234LS	AI-68K	¥188,000	AVCフタバ特価
CZ-620H	20MBハードディスク	¥178,000	AVCフタバ特価
CZ-64H	40MBハードディスク	¥120,000	AVCフタバ特価
LHD-34V	40MBハードディスク(ロジテック)	¥153,000	¥117,000
LHD-32V	20MBハードディスク(ロジテック)	¥128,000	¥98,000

CZ-8NJ2

アナログジョイステック
標準価格 ¥23,800

AVC特価¥???

CZ-8PG2

24ピンカラー。漢字ドットインパクトプリンタ

CZ-8PG2……¥160,000

AVC特価¥???

CZ-8PC4

48ドット熱転写プリンター。精密な文字、ハードコピーも可能。

CZ-8PC4……¥99,800

AVC特価 ¥64,800

CZ-8PC3

24ドット熱転写カラープリンター

標準価格……¥65,800

AVC特価 ¥39,800

●頭金なし(手軽な電話クレジット)●製品先取り(お支払いは約1~2ヶ月後から)●低金利クレジット(1回の支払いは2,700円以上で3~48回。ボーナス併用も可)●カレッジクレジット(保証人なし。但し満20歳以上の学生の方)●18歳未満の方(ご両親が代理購入者としてお申し込み下さい)●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい)●完全保証(すべてメーカー保証書付。アフターケア万全)●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時まで受付 日曜・祝日も営業

●但し消費税(3%)は別途請求させていただきます。●分割回数は3回~48回まで自由に選べます。

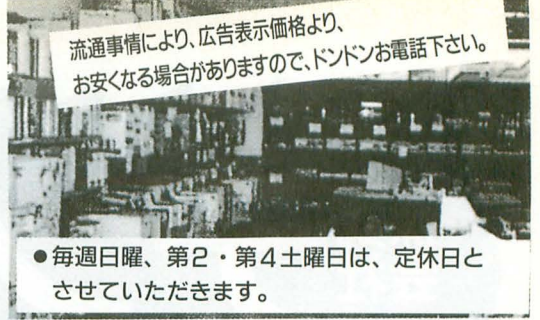
パソコン専門 O.A.ランド

アフターサービス万全
のサポート体制
優良パソコン販売店

●お近くの方は、お立寄り下さい。
専門係員がアドバイスいたします。
●ビジネスソフト、ゲームソフトのこと
ならおまかせ下さい!!

セール期間
◀ '90 6・15➡7・15

もうすぐ夏だヨ!! ドカ～ンとプレゼント
OAランド恒例・大お買徳セール実施中

●毎週日曜、第2・第4土曜日は、定休日とさせていただきます。

SHARP X68000シリーズセット (お楽しみゲームパック付)

●次代のインテリジェンス＝SX-WINDOW搭載!!

X68000 EXPERTII
●CZ-603C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計￥453,000
OAランド大特価

クレジット	12回	￥30,200	24回	￥15,900

NEW

X68000 EXPERTII-HD
●CZ-613C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計￥563,000
OAランド大特価

クレジット	12回	￥37,400	24回	￥19,700

●SX-WINDOW搭載!!

X68000 PROII
●CZ-653C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計￥400,000
OAランド大特価

クレジット	12回	￥26,600	24回	￥14,000

NEW

X68000 PROII-HD
●CZ-663C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計￥510,000
OAランド大特価

クレジット	12回	￥33,800	24回	￥17,800

●SX-WINDOW搭載!!

X68000 SUPER-HD
●SX-WINDOW搭載
●SCSIインターフェース装備
●80MBハードディスク搭載
●3MB大容量メモリ装備
●高解像度グラフィック

クレジット	12回	￥40,600	24回	￥21,400

NEW

X68000 SUPER-HD
●CZ-623C-TN(チタン)
●CZ-613D-TN(チタン)
●MD-2HD 20枚
定価合計￥633,000
OAランド大特価

X-1ターボZIIIセット

安すぎてゴメンなさい!

Ⓐセット
●CZ-888CBK…定価￥169,800
●CZ-880DBK‥ 定価￥109,800
●CZ-6ST1B…定価￥ 5,800
(チルトスタンド)
●MD-2HD 20枚サービス
合計定価￥275,400
特価中TEL下さい

Ⓑセット
●CZ-888CBK…定価￥169,800
●CZ-830DBK…定価￥ 98,000
●CZ-6ST-1B…定価￥ 5,800
(チルトスタンド)
●MD-2HD 20枚サービス
合計価格￥273,600
特価中TEL下さい

今月の特価品(限定)お早目に!!

★CZ-652C(BK)＋CZ-602D(BK)
4セット限り……大特価￥258,000

★CZ-612C(BK)
3セット限り…大特価￥298,000

●SHARP WD-A300(ワープロ)
定価￥165,000……特価￥115,000
●SHARP WD-A330(ワープロ)
定価￥185,000……特価￥129,000
●SHARP WD-HL30(ワープロ)
定価￥198,000……特価￥138,000
●SHARP PW-910(ワープロ)
……特価￥ 85,000

●NEC PC-KD853(アナログCRT)
……特価￥ 50,000
●三菱XC-1498C(アナログCRT)
……特価￥ 54,800
●SHARP CU-14FD(アナログCRT)
……特価￥ 46,000
●SHARP PA-8500(電子手帳)
……特価￥ 16,800

周辺機器コーナー

プリンターセットコーナー

●CZ-6PV1(カラービデオプリンター)
定価￥198,000……▶特価￥152,000
●CZ-8PC3(24ドット熱転写カラープリンター)
定価￥ 65,800……▶特価￥ 53,000
●CZ-8PK10(24ピン漢字ドットプリンター・136桁)
定価￥ 97,800……▶特価!TEL下さい!
●CZ-8PG1(24ピンカラー漢字ドットプリンター・80桁)
定価￥130,000……▶特価!TEL下さい!
●CZ-8PG2(24ピンカラー漢字ドットプリンター・136桁)
定価￥160,000……▶特価!TEL下さい!
●IO-735X(カラーイメージェットプリンター)
定価￥248,000……▶特価!TEL下さい!

OAランド特選品!!

■CZ-8PC4(定価￥99,800)
●48ドット熱転写カラー漢字プリンター
特価￥64,800

X68000用ソフトウェアー・コーナー

	品名	定価	特価
①	CZ-212BS(BUSINESS)	￥68,000	￥53,000
②	CZ-220BS(DATA)	￥58,000	￥45,000
③	CZ-215MS(Sampling)	￥17,800	￥13,800
④	CZ-221HS(NEW Print Shop)	￥10,800	￥15,500
⑤	CZ-227BS(TOP財務会計)	￥200,000	￥158,000
⑥	CZ-226BS(CARD)	￥229,800	￥23,000
⑦	CZ-223CS(Communication)	￥19,800	￥115,500
⑧	CZ-213MS(MUSIC)	￥18,800	￥14,800
⑨	CZ-211LS(C compiler)	￥39,800	￥31,000
⑩	C-TRACE(キャスト)	￥68,000	￥52,000
⑪	EW(イースト)	￥38,000	￥29,000

X68000用周辺機器コーナー

品名	定価	特価
CZ-6PU1A	￥38.000	￥30,000
CZ-6BM1	￥26.800	￥21,000
CZ-6BE1	￥88.000	￥69,800
CZ-6VT1	￥69.800	TEL下さい
CZ-8NS1	￥188.000	￥149,000
CZ-6BC1	￥79.800	￥63,000

●最新ゲームソフト
その他各種ソフト
20%～25%OFF!!
●周辺機器・プリンター
割引販売中!! TEL下さい!

■I・O DATA 増設RAMボード NEW

●1MB増設RAMボード
PIO-6BE1-A
定価￥25,000
特価￥19,500

●2MB増設RAMボード
PIO-6BE2-2M
定価￥50,000

特価￥38,500

●4MB増設RAMボード
PIO-6BE4-4M
定価￥88,000

特価￥67,000

■ハードディスク ■特価品もありますので TEL下さい。

●アイテック ITX-640……特価￥117,000
●アイテック ITX-680……特価￥149,000
●ロジテック LHD-32V……特価￥ 85,000
●ロジテック LHD-34VE……特価￥ 90,000
●ロジテック LHD-34V……特価￥104,000
●シャープ CZ-620H……特価￥118,000
●シャープ CZ-64H……特価￥ 95,000
●アイテム HXD-040……特価￥ 88,000
●アイテム HXD-042……特価￥ 95,000
●ICM SR-80……特価￥130,000

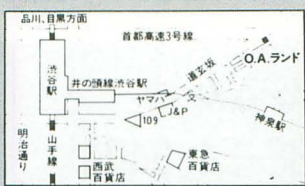
中古パソコン (価格/在庫は変動します。予約は5日以内とします。)

機種	価格	機種	価格
PC-9801RA5	￥338,000より	PC-286VS	￥165,000より
PC-9801RA2	￥265,000より	CZ-600C	￥160,000より
PC-9801RX2	￥199,000より	CZ-601C	￥170,000より
PC-9801EX2	￥190,000より	CZ-611C	￥198,000より
PC-9801VX21	￥170,000より	CZ-652C	￥178,000より
PC-9801UX21	￥165,000より	CZ-612C	￥210,000より
PC-9801VX2	￥160,000より	68000用モニター	￥ 49,000より
PC-9801VM21	￥150,000より	PC-9801用サウンドボード	￥ 13,000より
PC-9801UV11	￥148,000より	PC-88SR,FR	￥ 50,000より
PC-9801LV22	￥160,000より	PC-88FH,FA	￥ 65,000より
PC-286VE	￥150,000より	400ラインCRT	￥ 38,000より
PC-286US	￥155,000より	200ラインCRT	￥ 10,000より

通信販売のご案内

全国通販

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド
■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1～60回払で月々5,000円よりご自由に設定できます。

品川、目黒方面
首都高速3号線
O.A.ランド
渋谷駅
井の頭線渋谷駅
道玄坂
ヤマハ
109
J&P
神泉駅
明治通り
山手線
西武百貨店
東急百貨店

●下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
●ご注文、お問合せは… 午前10時から午後7時まで
●商品のお届けは…入金確認後、即日発送致します。

㈱オー・エー・ランド

〒150東京都渋谷区円山町20-4 第5日新ビル1F
☎(03)770-8855 FAX(03)770-7080
関東エリアの送料は、1個につき￥1,000です。
★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、5月末現在です。

株式会社 デンキヤ

営業時間AM11:00〜PM7:00 水・木曜定休

（価格は全べて税込みです）

セット超特価

X68000
PERSONAL WORKSTATION

PROⅡ・PROⅡ HD

CZ-653C特価

CZ-663C特価

SUPER HD

CZ-623C特価

CZ-613D特価

セット超特価

X68000
PERSONAL WORKSTATION

EXPERTⅡ・EXPERTⅡ HD

CZ-603C特価

CZ-613C特価

EXPERT PRO

CZ-662C特価

CZ-602C特価

全品メーカー保証 即決クレジットOK

ディスプレイ		プリンタ		周辺機器		ソフト	
CZ-604D	特価	CZ-8PC4	特価	CZ-8NJ1	¥1,400	CZ-213MS	¥15,500
CZ-605D	特価	CZ-8PG1	特価	CZ-8NJ2	¥18,540	CZ-223CS	¥15,300
CZ-613D	特価	CZ-8PG2	特価	PIO-6BE1A	¥20,000	CZ-219SS	¥23,100
CU-21HD	特価	IO-735X	特価	PIO-6BE2	¥39,000	CZ-211LS	¥30,800

24時間テレホンサービス

0482-54-3444

お申し込み

TEL.0482-54-3400

FAX.0482-54-3443

埼玉県川口市西川口4-6-4

お支払い

下記取引銀行口座

までお振込み下さい。

三菱銀行西川口支店

㈱デンキヤ㊪0258081

アイ・ツー EXE CLUB

新規ユーザー・EXE会員大集合

★X68000ユーザーニーズに対応したハード・ソフト・ウエア・周辺機器は全て展示しています。
★新製品情報・ユーザー同士の情報交換ができる、メンバー様の憩いのスペースです。
★大特価セール期間中X68000・ディスプレイ・プリンター御購入の方は全国どこでも送料無料!!
★遠くでなかなかお越し頂けない方にも通販専用TELで専門スタッフ(X68 PRO STAFF)が親切丁寧にお答えします。
★X68000お買い上げの方、アイツーよりBigプレゼント。

X68000 オリジナルステッカー
X68000 フロッピータイトルシール
X68000 オリジナルテレフォンカード
X68000 バッグ
お好きなもの2点もれなくついてくる!!

★現在シャープX68000 EXE会員の方、おトモダチをご紹介下さい。ご購入成立時点でアイ・ツーとシャープよりステキなプレゼント進呈中!!
★アイ・ツーメンバーズ優待制度実施
アイ・ツーでX68000・及びソフトウエア周辺機器をお買上げ頂きましたユーザー様にはオリジナルメンバーズカードを送付致します。メンバーズの方には楽しいパソコンライフをおくれますように最善のフォローをアイ・ツーより提供します。

春は気分も新たにいろんなことをやってみたい。そんなあなたにとっておきの贈り物をシャープから。

★X68000新製品
Super-HD(CZ-623C)、EXPERTⅡ(CZ-603C)
EXPERTⅡ-HD(CZ-613C)、PROⅡ(CZ-653C)
PROⅡ-HD(CZ-663C)　ゾクゾク入荷!!
X68000のことなら、なんでもご相談下さい。
旧型製品も格安にて"御提供中"

NEW X68000誕生
PERSONAL WORKSTATION
SUPER・EXPERT・PRO
●SX-WINDOW搭載 オリジナルウインドゥシステム
●大容量メディア対応 SCSI対応、大容量H.D.D
●AV機能強化 ビデオボード、音多・AVディスプレイテレビ
頂点を極める3大機能

SB会販売コンテスト・パソコン部門最高峰賞
シャープ販売第一位受賞感謝セール!
期間／6月18日～7月17日

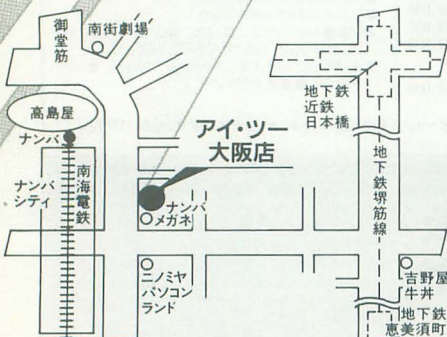
FAX特集
SHARP FO-50 定価¥ 99,800
SHARP UX-10 定価¥128,000
SHARP UX-20 定価¥168,000
大特価にて展示中!
只今お買上の方、絶対必要なロール紙プレゼント中!!
全国どこでも送料無料。カタログ希望の方、ハガキに御使用機種(CZ…とか)ご住所、ご氏名、TEL、生年月日をお書きのうえアイ・ツーEXE CLUBへご送付下さい!

御礼
アイ・ツーinシャープグランドフェア'90 OSAKAスタジアムに多数のご来場頂きまして、誠にありがとうございました。アイ・ツーサンクスフェアPart2も只今企画中ですので、迄うご期待!!

X68000プロショップ(専門店)ならではの企画です!!

パソコンミュージック MIDI X68000
音遊サウンドドライブ　店頭イベント
X68000"MIDI"実体験
(パソコンができなくても、楽器ができなくても"MIDI"ができる)
店頭では"MIDI"だよX68000を切り口にパソコンミュージック"MIDI"が体験できる「X68000トレンド実体験コーナー」を設置X68000新製品(SUPER HD・EXPERTⅡ・PROⅡ)ラインアップ展示でX68000の魅力の全てを訴求します。

逃がす手はないチャンスです

X68000ユーザーとっておきのグッズ!!
X68000ユーザーのステータスシンボル。
新グッズもグループインしてますます充実。
キミのパソコンライフが一層楽しくなるコレクションだ!
X68000オリジナルグッズをまだ持っていないキミ
アイ・ツーからお届けしちゃいマス!

■営業時間 AM11:00～PM8:00

通販専用TEL.
06-633-9800

年中無休
Information & Interface
株式会社アイ・ツー
大阪店／〒542 大阪市中央区難波千日前15-18

クリエイト特典

- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00〜PM7:00

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

X68000 ボーナスセール開催中!!

X68000 NEW PROII

- ●CZ-653C(本体)……………………¥285,000
- ●CZ-603D(カラーディスプレイ)……………………¥ 84,800
- ●お好きなゲームソフト1本……………………¥ 7,800
- ■定価合計……………………¥377,600

クリエイト特価

均等払い	¥ 7,680×48回	¥ 9,890×36回	¥14,370×24回
ボーナス	な し	な し	な し

X68000 NEW EXPERTII

- ●CZ-603C(本体)……………………¥338,000
- ●CZ-613D(カラーディスプレイテレビ)……………………¥ 99,800
- ●CZ-8NJ2……………………¥ 23,800
- ●お好きなゲームソフト1本……………………¥ 9,800
- ■定価合計……………………¥506,600

クリエイト特価

均等払い	¥ 9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	な し	な し	な し

X68000 EXPERTII HD

- ●CZ-613C(本体)……………………¥448,000
- ●CZ-604D(カラーディスプレイ)……………………¥ 94,800
- ●お好きなゲームソフト1本……………………¥ 9,800
- ■定価合計……………………¥552,600

クリエイト特価

均等払い	¥ 5,920×48回	¥ 7,400×36回	¥12,100×24回
ボーナス	¥30,000× 8回	¥40,000× 6回	¥50,000× 4回

X68000 SUPER HD

- ●CZ-623C-TN(本体・キーボード・マウス)……………………¥498,000
- ●CZ-613D-TN(カラーディスプレイ)……………………¥135,000
- ●CZ-6BP1……………………¥ 79,800
- ■定価合計……………………¥712,800

クリエイト特価

均等払い	¥ 7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000× 8回	¥50,000× 6回	¥80,000× 4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERTII

ミュージシャンセット。これもTMネットワークだよ〜/

- ●CZ-603C……………………¥338,000
- ●CZ-605D……………………¥115,000
- ●MU1.B(MIDIボード&ソフト)……………………¥ 39,800
- ●CM32L……………………¥ 69,000
- ●グラナダ……………………¥ 8,800
- ●JOYカード……………………¥ 1,800
- ■定価合計…………¥572,400▶超特価¥458,000

X68000 NEW PROII

ゲーマーズセット。遊んで暮らせるSET/

- ●PROII CZ653C……………………¥285,000
- ●0.31CRT CZ603D……………………¥ 84,800
- ●グラナダ……………………¥ 8,800
- ●Y'S……………………¥ 8,700
- ●ポピュラス……………………¥ 9,800
- ●スーパーハングオン……………………¥ 8,800
- ●エージャックス……………………¥ 8,800
- ●サーク……………………¥ 8,800
- ●アールタイプ……………………¥ 7,800
- ●アナログJOYSTIC XE-1AP……………………¥ 13,800
- ■定価合計…………¥445,100▶超特価¥353,000

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型 番	品 名	定 価	ソフト名	品 名	定 価
CZ-6VT1	カラーイメージユニット	¥ 69,800	MUSIC PRO	MIDI版	¥ 28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥ 18,800
CZ-6BE1A	IMB増設RAMボード	¥ 38,000	SOUND PRO-68K	サウンドエディタ	¥ 15,800
CZ-6BE2	2MB増設RAMボード	¥ 79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥ 17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥ 28,800
CZ-8NM3	マウス・トラックボール	¥ 9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥ 29,800
BF-68PRO	高性能CRTフィルター	¥ 19,800	PRO-68K	サイバーノート	¥ 19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800	PRO-68K	ステーショナリー	¥ 14,800
CZ-8NT1	トラックボール	¥ 13,800	Ccompiler PRO-68K	ソフト開発セット	¥ 39,800
CZ-6BM1	MIDIボード	¥ 26,800	Human 68K Ver2.0	開発ツールセット	¥ 9,800
CZ-8NJ2	アナログスティック	¥ 23,800	PIO-6BE1-A	内蔵1MRAM	¥ 25,000
CZ-6TU	パソコンチューナ	¥ 33,100	PIO-6BE2-2M	2MRAM	¥ 50,000
SX-68M	MIDI I/F	¥ 19,800	PIO-6BE4-4M	4MRAM	¥ 88,000
XE-1AP	アナログジョイパッド	¥ 13,800	MU1-B	MIDI I/F+ソフト	¥ 39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。 ●超特価販売中/

オール15%〜20%OFF

パソコン専門ショップ

総合お問合せ先☎03-486-6541(代)

ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店(普)No.5000340

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

こんどの 6月24日(日)

MIDIって何だ？ というあなたも、コンピューターミュージックならまかせろ！という君も、思わずナットク。スタジオでの多重録音による音楽作成の現場をプロミュージシャンがトークショー形式ですべてお見せします。こんなチャンスはざらにない！

ADO·TOYOMURA T·ZONE ティー・ゾーン

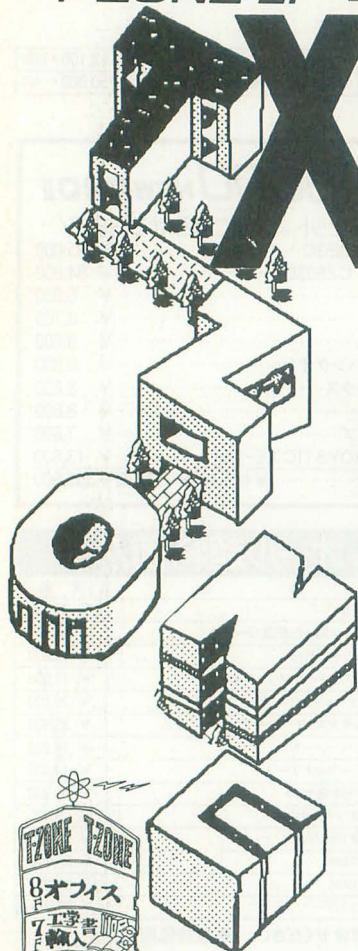
Micom Zone

2F 〒101 東京都千代田区外神田4-4-1 ☎257-2650

海外でも使える
「T·ZONE CLUB」
カード会員募集中!!
「オリエント」「UC」「マスター」カードが１つになった。
「ボーナス一括払い」OK！「通信販売」も
お手軽にご利用頂けます。そのほか、便利でお得な特典がいっぱい！ 今がチャンス!!
詳しくは、店頭にてどうぞ!!

X68000をトータルサポート
T·ZONE 2F

SHARP Authorized………

X68000 PRO SHOP

X68000
新ラインナップ大好評発売中!!
EXPERTⅡ·EXPERTⅡ HD
PROⅡ·PROⅡ HD · SUPER HD

■新高速BIOS、SX-Window等、期待のX68000新シリーズを全品超特価販売中です。
もちろんクレジットもOK！
X68000シリーズはT·ZONEにおまかせ下さい。

OS-9／X68000

- □OS9/68000 (SHARP) ¥29,800
- □C & PRO PACK (マイクロウェア) ¥58,000
- □MW-BASIC (マイクロウェア) ¥60,000
- □BTree09 (ARK) ¥36,000
  MW-BASIC用のISAM用B-Treeパッケージです。応用例として住所録と販売管理プログラムが付属。全ソースコード付です。(このソフトを動かすためにはMW-BASICが必要です。)
- □UD-CACHE (ARK) ¥16,000
  すべてのRBFデバイスに対応するキャッシュです。
- □FBU (ARK) ¥38,000
  ハード・ディスクバックアップユーティリティーです。巨大ファイルを分割バックアップしたり、日付管理を行なったバックアップもOK。
- □VSED (FORKS) ¥28,000
  OS9/68000で唯一オートバッファリングをサポートしたスクリーンエディタです。

ニューリリースソフトウェア

CSG IMS V2.0
Clearbrook Software Group Information Management System
㈱星光電子
¥118,000

あのCSG-IMSがついにX68000にリリースされました。高度の処理に対応可能な言語型リレーショナルデータベースです。フォーマッタを利用して簡単にシステム設計を行えます。OS9の特長を活かして、リアルタイム、マルチタスク、マルチユーザが必要なアプリケーションを構築することが可能です。もちろん、Cやアセンブラのモジュールを呼び出すことも可能、OAはもちろんFAにも対応できるパワフルツールです。

OS9-SHL　FORKS
Super Shell for OS9/68K ¥12,800
お待ちかね、OS9-SHLがX68000に対応しました。
- ■プロシージャファイルで引数、リダイレクト、パイプが使用可能。
- ■環境変数を数値または文字列として演算可、制御文中で使用できます。
- ■豊富な制御文。While, wend, loopout, continue, if, else……
- ■ヒストリやエイリアスをサポート、マクロコマンドもOK。
- ■カーソルキーに対応。

C SRC DBG マイクロウェアシステムズ
C SOURCE LEVEL DEBUGGER ¥39,800
OS9/X68000用Cソースレベルデバッガ発売開始!!
これでデバッグも楽々。

番外 Oh! FMコーナー
FMシリーズ用OS9に新ソフト登場！
日本ソフトバンク
DB-09 (FM-7, 77, AV, 11) ¥18,252！
OS9上で走るリレーショナルデータベースマネージャーです。問い合わせ形式で取扱い簡単。なんとCによる全ソース付。

T·ZONE 正社員・長期アルバイト募集中！
☆お問い合わせは総務課鈴木まで(TEL 03-257-2630)

T·ZONE T·ZONE
8F オフィス
7F 工学書 輸入
6F 計測器 衛星放送
5F HAM トランシーバー
4F トレード ゾーン
3F IBM ゾーン
2F パソコン ソフト ウェア
1F パーソナル OA
B1 Macintosh!

T·ZONE
営業時間：AM10:30〜PM7:00

下記T·ZONE各店でも扱っています。
宇都宮店：☎0286(63)4949 川口店：☎0482(68)7826 静岡店：☎0542(83)1331 横浜店：☎045(641)7741
大宮店：☎048(652)1831 東ラジ店：☎03(257)2694 パーツショップ：☎03(257)2655

●マイコン通販利用の方へ：現金書留て送金される際は、住所、氏名、TEL番号、希望商品名(詳しく)を明記して下さい 振込を御希望の方は下記銀行へお願いします
尚、いずれも予めTELにて、御予約・送料確認の上御送金下さい (振込口座 埼玉銀行 秋葉原支店 当座2705 ㈱亜土電子工業)

☆この広告の提示価格には、消費税は含まれておりません。

T4910217907565 雑誌02179−7